

RTE-IVB QUICK REFERENCE GUIDE



**HEWLETT
PACKARD**

**DATA SYSTEMS DIVISION
11000 WOLFE ROAD
CUPERTINO, CALIFORNIA 95014**

**Library Index Number
2RTE.340.92068-90003**

**MANUAL PART NO. 92068-90003
Printed in U.S.A. July 1980**

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages contain replacement pages or write-in instructions to be merged into the manual by the customer. Manuals will be reprinted as necessary to incorporate all prior updates. A reprinted manual is identical in content (but not in appearance) to the previous edition with all updates incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

First Edition Jan 1980
Second Edition July 1980

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1980 by HEWLETT-PACKARD COMPANY

TABLE OF CONTENTS

	SECTION
SYSTEM AND BREAKMODE COMMANDS	A
FMGR COMMANDS	B
BATCH AND SPOOLING COMMANDS	C
GASP COMMANDS	D
ACCOUNTS COMMANDS	E
EDITR COMMANDS	F
UTILITIES	G
EXEC CALLS	H
FMP CALLS	I
SMP CALLS	J
TABLES	K
ERRORS	L

SYSTEM AND BREAKMODE COMMANDS

CONTENT	PAGE
AB	A-2
AS	A-2
BL	A-2
BR	A-2
DN	A-2
EN	A-3
EQ	A-3
FL	A-3
GO	A-3
HE	A-4
IT	A-4
LU	A-4
OF	A-4
ON	A-5
OP	A-5
PR	A-5
QU	A-5
RS	A-5
RT	A-5
RU	A-5
SL	A-6
SS	A-6
ST	A-6
SZ	A-7
TE	A-7
TI	A-7
TM	A-7
TO	A-7
UP	A-8
UR	A-8
WH	A-8

SYSTEM AND BREAKMODE

AB,optn --

Abort currently executing batch job. Under session, the command is valid only when entered from the system console.

- optn 0 Disc tracks not released.
- 1 Release all disc tracks.

AS,program,partition# 50

Assign a program to always execute in same partition. To unassign, set partition = 0.

BL 10

Examine current buffer limits

BL[,lower[,upper]] 60

Modify current buffer limits.

- lower Limit specified in number of words (default=0).
- upper Limit specified in number of words (default=existing limit).

BR[,program] 10/60

Set break flag for any program in user's session. User programs tests for a set break flag with subfunction I=IFBRK (DUMMY). Required capability (Default=current session program.)

Set break flag in any program in the system. Requires capability of 60.

DN,,lu 60

Set I/O device down.

- lu system logical unit.

DN,eqt 60

Set I/O controller down.

- eqt equipment table entry number.

EN,mstr scty code[,option]

--

Enable system console as a session terminal. Command only valid when entered from the system console.

mstr scty code Two character FMP master security code.

option 0 master security code not required in "OP" commands (default).
1 master security code is required in "OP" commands.

EQ,eqt

10

Print description and status of an I/O controller. Status information is printed as.

select code DV.nn D B Unn status

select code is the I/O select code number.

DV.nn is the driver routine.

D is D if DMA required; 0 if not.

Unn is B if automatic output buffering; 0 if not.

status is the logical status:
0 = available.
1 = I/O controller down.
2 = I/O controller busy.
3 = waiting for DMA assignment.

UNbuffer

EQ,eqt,

BUffer

60

Change the automatic buffering designation for a particular I/O device.

FL

10

Eliminate buffered output to a session terminal. Only valid in break mode, and not valid from system console.

GO[IH][,program][,pl[,...[,p5]]]]

30/60

Reschedule any program in users session, where parameters are passed by program only when it has suspended itself. GOIH inhibits passing of command string. Requires capability of 30.

Reschedule any program in the system. Requires capability of 60.

SYSTEM AND BREAKMODE

HE[,keyword[,lu]] 1

Detailed error explanation.

- keyword** an eight character error code (default=last error logged).
- lu** device for explanation (default=user's terminal).

IT,program[,res,mpt[,hr,min[,sec[,tms]]]] 50

Set automatic execution time value for a program. ON command must follow to schedule the program. Not specifying optional parameters removes "program" from the timelist (program must be dormant).

- res** resolution code:
 - 1 tens of ms
 - 2 seconds
 - 3 minutes
 - 4 hours
- mpt** multiplier (0-4095) used with res.
- hr,min** Initial start time.
- sec,tms**

LU,lu 60

Print EQT entry number, device subchannel number, associated with a system lu, and whether the device is up or down. See SL command for similar function.

LU,lu,0 60

Reassign system lu to be bit bucket.

LU,lu,eqt[,subchannel #] 60

Reassign new EQT entry number to system lu. If EQT number has subchannels, use subchannel #.

OF,program[,numb] 30/60

Terminate a session program. Requires capability of 30.
Terminate any program in the system. Requires capability of 60.

- numb** 0 remove from time list; disc tracks not released (default).
- 1 terminate immediately; release disc tracks
- 8 terminate immediately and permanently from system (must be issued to segments as well as the main).

ON[IH],program[,NOW][,parameters]

50

Schedule a program for execution. Program's entry in time list is affected. ONIH inhibits passing of command string.

NOW Schedule program immediately.

parameters 1-5 parameters passed to program when it is scheduled.

OP[,mstr scty code[,command]]

--

Enter a system level command from a low capability session. Command only valid when entered from the system console.

mstr scty code Two character FMP master security code. If specified in the "EN" command the security code is required.

command The system command to be executed.

PR,program,priority

50

Change program priority where priority = 1-32767 (decimal).

QU[,quantum[,limit]]

10/60

Examine system timeslice quantum and fence. Requires capability of 10.

Modify system timeslice quantum and fence. Requires capability of 60.

quantum system timeslice quantum, value 0-32767 millisecs (default=1500).

limit priority level fence to begin timeslicing (default=50).

RS

10

Abort and reschedule a session's copy of FMGR.

RT,program

30

Release all disc tracks assigned to a program.

RU[IH],program[,parameters]

30

Schedule a program for immediate execution. Program's entry in time list is not affected. 1-5 parameters are optionally passed to program when it is scheduled. RUIH inhibits saving of command string. The breakmode RU actually runs "program" not a renamed copy of "program".

SL[,lu]

10

Display session lu information.

lu session lu for which linkage information is desired.
 (Default=information for all session lu's in user's
 session switch table.)

SS[,program]

30/60

Suspend non-dormant session program. Requires capability of 30. If
program name not specified, the current session program is
suspended.

Suspend non-dormant system program. Requires capability of 60.

ST,name

10

Determine status of named program. Status is printed as:

pr S res mpt hr min sec ms T

pr Decimal priority.

S current state of program:

 0 Dormant

 1 Scheduled

 2 I/O suspend

 3 General wait

 4 Unavailable memory suspend

 5 Disc allocation suspend

 6 SS or EXEC 7 suspend

 9 Background segment

res/mpt/ 0 or

hr/min/sec time program is next scheduled to run.

/tms

T Program currently in time list.

ST[,numb]

10

Determine name or partition number of program currently executing.

numb 0 — Display name and partition number of pro-
 gram currently executing in memory. 0 dis-
 played if none executing.

 Partition # — Display name of program currently
 residing in that partition. 0 if none.

SZ,program

30

Display the named program's size information as follows:

AAAAA BB CCCC DD

AAAAA last word plus 1 of program.

BB required partition size. Program code + EMA.

CCCC EMA size (EMA programs only).

DD MSEG size (EMA programs only).

SZ,program,size[,MSEG size]

30

Change size of "program".

program program name.

size Non-EMA program: required program size.
EMA program: required EMA size.

MSEG size new MSEG size (EMA program only).

TE,message

10

Send message to system console.

TI

10

Print current year, Julian day and time.

TM,year,day[,hr[,min[,sec]]]

60

Set real time clock.

year four digits (e.g., 1957).

day three digits Julian date (e.g., 063 = March 4).

TO,eqt[,numb]

10/60

Examine device time out parameters. Requires capability of 10.

Change device time out parameters. Where numb is number of 10 ms intervals used as new time out value. Requires capability of 60.

SYSTEM AND BREAKMODE

UP,eqt 10

Make I/O controller (and all associated lu's) available.

UR,partition # 50

Release reserved partition.

WH[,lu[,option]]
or 10
WH[,option]

Schedule WHZAT program.

- | | |
|--------|---|
| lu | the session lu for display. (default=user's terminal). |
| option | default User's session programs. |
| AL | Display status of all suspended and scheduled programs. |
| SM | Similar to AL except, state 3 programs without father son relationships are not listed. |
| PA | Display status of all partitions. |

FMGR COMMANDS

CONTENT	PAGE
AC	B-4
AN	B-4
CA	B-4
CL	B-4
CN	B-5
CO	B-5
CR	B-5
CS	B-6
CT	B-7
DC	B-7
DL	B-7
DP	B-8
DU	B-8
EX	B-8
HE	B-9
IF	B-9
IN	B-9
LI	B-10
LL	B-10
LO	B-10
MC	B-10
ME	B-11
OF	B-11
PA	B-11
PK	B-11
PU	B-11
RN	B-11
RP	B-12
RT	B-12
RU	B-12
SE	B-12

CONTENT	PAGE
SL	B-12
SM	B-14
SP	B-14
ST	B-15
SV	B-15
SY	B-16
TE	B-16
TR	B-16
WH	B-16
??	B-16
*	B-16

PARAMETERS

namr=name[:security[:cartridge
[:type[:file size[:record size]]]]]

or

namr=logical unit number

security	<0 Write and read protected 0 Not protected (default) >0 Write protected
cartridge	<0 lu number 0 First available cartridge (default) >0 FMGR cartridge reference number
file type	0 Non-disc file 1 128-word record length, random access 2 User selected record length, random access 3 (and greater) variable record length, sequential access 4 Source program 5 Relocatable program 6 RTE load module 7 Absolute program >7 User defined
file size	Specified in blocks (2 sectors = 1 block = 128 words). +n = allocate n blocks. -n = allocate n 128 block multiples. -1 = allocate remaining space on cartridge.
record size	Used only when file is type 2.

SCHEDULING FMGR

RU,FMGR[,namr[,list[,severity code[,log]]]]

namr	File name or lu containing command input.
log	lu of log device (default=input or LU1).
list	lu of list device (default=LU1).
severity code	Display commands and error codes. 0 Display all commands and errors (default). 1 Display no commands, all errors. 2 Display no commands, no errors except those requiring response. Terminates job on serious error. 3 Same as 2 except job not terminated. 4 Display no commands, no errors, and do not abort job.

AC,crn[,P/G[,size[,id[,# dir. tracks]]]]

10

Allocate a cartridge to the session user from the spare cartridge pool,

crn	Cartridge reference number to be assigned to the allocated cartridge.
P/G	Private (P) or group (G) cartridge designation (default=P).
size	Number of tracks needed on cartridge.
id	ASCII identifier of cartridge (default=DC00XX;XX is system lu number of terminal).
#dir. tracks	# of tracks used by file directory (default=1).

AN,message

20

Print message on list device.

CA,global#[,pl[op1,p2[... ,op(n),p(n + 1)]]]

40

Calculate global parameter values.

global#	Integer preceding G in G-type global, or "integer:P" for P-type globals.
pl-pn	Values used in calculations; if omitted, global is nulled.
op1-opn	Operations performed on operands pl-pn. + add two operands - subtract second operand from first / divide second operand by first * multiply two operands O OR X XOR (exclusive OR) A AND

CL[AL]

10

Display list of user accessible cartridges.

AL	Display list of all cartridges in system.
----	---

CN[,namr[,function[,subfnctn]]]

20

Issue control request to non-disc device.

namr	Type 0 file name or lu (default=LU8).																
function	Control code, mnemonic (for octal see EXEC 3 call). mnemonic <table> <tr><td>RW</td><td>rewind (default=MT,CTU)</td></tr> <tr><td>EO</td><td>end-of-file</td></tr> <tr><td>TO</td><td>top-of-form (default=LP,CRT)</td></tr> <tr><td>FF</td><td>forward space file</td></tr> <tr><td>BF</td><td>backspace file</td></tr> <tr><td>FR</td><td>forward space record</td></tr> <tr><td>BR</td><td>backspace record</td></tr> <tr><td>LE</td><td>leader (default=paper tape punch)</td></tr> </table>	RW	rewind (default=MT,CTU)	EO	end-of-file	TO	top-of-form (default=LP,CRT)	FF	forward space file	BF	backspace file	FR	forward space record	BR	backspace record	LE	leader (default=paper tape punch)
RW	rewind (default=MT,CTU)																
EO	end-of-file																
TO	top-of-form (default=LP,CRT)																
FF	forward space file																
BF	backspace file																
FR	forward space record																
BR	backspace record																
LE	leader (default=paper tape punch)																
subfnctn	Carriage control. <table> <tr><td>+n</td><td>to space n lines before next print operation.</td></tr> <tr><td>-n</td><td>page eject on line printer or space -n lines on terminal.</td></tr> </table>	+n	to space n lines before next print operation.	-n	page eject on line printer or space -n lines on terminal.												
+n	to space n lines before next print operation.																
-n	page eject on line printer or space -n lines on terminal.																

CO,cartridge1,cartridge2

20

Copy all files from active cartridge 1 to active cartridge 2.

CR,namr

20

Create a disc file — data not transferred, namr subparameters required:

file type (must not be 0).
 file size (must not be 0).
 record size (when type=2).

CR,namr,lu,REad,BSpace,EOf,Blnary
WRite,FSpace,LEader,AScii
BOTH,BOTH,Page,cntrl
,cntrl

Create a non-disc (type 0) file — data not transferred.

namr	File name, security code, and crn.
lu	Lu of non-disc device (positive).
REad	
WRite	Legal input/output (no default).
BOTH	
BSpace	
FSpace	Legal spacing (default=FS for READ devices, no
BOTH	space all others).
EOf	
LEader	Control subfunction (default=EO for mass storage
Page	devices, LE for paper tape punch, PA for line
cntrl	printer).
Blnary	
AScii	Type of data (default=AS).
cntrl	

CS,lu,attribute

Modify or change spool options set up by SL command.

lu	Lu defined at set up.
attribute	One of the following:
RWind	reset file to first record
PUrge	change SAve flag to PUrge
SAve	change PUrge flag to SAve
PAss	remove HOld option
ENd	write EOF and terminate spool. Spool file
	placed in outspool queue (default).
BUffer	change to buffering
NBUffer	change to no buffering
NPass	change lu and/or priority information, by
	specifying the 2 additional parameters:
	[,outlu[,priority]]
	outlu = new lu.
	priority = new priority.

CT,name[,function[,subfnctn[,message]]]

10

Issue control request to terminal.

name	Type 0 file or terminal lu number.
function/	Octal code:
subfnctn	11B Space down a specified number of lines. subfunction: 0 skip 2 lines. +n skip n lines. -n skip n lines. 20B Enable terminal (default) 21B Disable terminal 22B Set time out. Subfunction: value in units of 10 msecs.
message	Message to be written to terminal.

DC,cartridge[,RR]

10

Logically remove a cartridge from session user's environment by setting inactive bit in session control block. Non-session, deletes entry in system cartridge list.

cartridge	Positive cartridge reference number or negative lu.
RR	Session only — deletes cartridge entry in system cartridge list.

[,cartridge[,security]]**DL or****,namr[,security]**

10

List the file directory of one or all of the mounted cartridges.

cartridge	Cartridge reference number, positive for label or negative for lu. Zero or none specified lists all.
namr	Mask specifying the file entries in the directory to be output. Minus signs (–) can be used as place holders for more flexibility.
security	Two-character FMP master security code.

If the master security code is 0, default in command will not obtain long list showing security codes — a code (any code) must be supplied.

FMGR

DP[,p1[,p2[,p3...[,pn]]]] 20

Display parameter value or global names. p1-pn are parameters to be displayed

DU,namr1,namr2[,record format[,file#[,#files]]] 20

Transfer data from an existing file or lu to another existing file or lu.
Does not create namr2.

namr1	Source of data
namr2	Destination of data
record format	Format of data or EOF control (default=namr1 format, or ASCII if non-disc device).
	ASCII ASCII records.
	BReloc Binary relocatable records with checksum.
	BNary Binary records without checksum.
	BAbs Binary absolute records with checksum.
	MTape Magnetic tape ASCII records.
	MS Magnetic tape SIO (System Input/Output) records are written on namr2. Standard records are expected on namr1.
	MSBR Magnetic tape SIO binary relocatable records (same as MS+BR).
	MSBA Magnetic tape SIO binary absolute records (same as MS+BA).
	IHibit Inhibits EOF on namr2 and leader punching.
	SAve Save embedded EOF's in namr1.
file#	File or subfile on namr2 where transfer starts (default=1).
#files	Number of files to be transferred from namr1 (default=1).

EX 1

Terminate FMGR.

SP
EX, [,RG[,KI]] 1
RP

Initiate log-off process.

SP/RP	Save/release private cartridges.
RG	Release group cartridges.
KI	Abort any active session programs.

HE[,keyword[,lu]]

1

Detailed error code explanation.

keyword	Identifiers related to error code (session default=last error posted). Non-session, keyword must be specified.
lu	Device for explanation output (default=user's terminal).

IF,p1,xx,p2[,skip]

40

Compare two values (usually globals) and skip a specified number of commands. Command not allowed from interactive device, must be in procedure file or batch job.

p1,p2	Values to be compared.
xx	ASCII operators as follows: EQ p1 = p2 NE p1 ≠ p2 LT p1 < p2 GT p1 > p2 GE p1 ≥ p2 LE p1 ≤ p2
skip	Number of commands to skip (positive or negative). Use -2 to skip back to previous command (default=1).

IN,mstr scty code,ctrldge,lbl,id[,1st trk[,#dir trks[,#sec/trk[,bad trks]]]]

60

Initialize a cartridge.

mstr sec code	Two-character FMP master security code.
ctrldge	Cartridge reference number, positive for label or negative lu. (Must be -lu if new.)
lbl	New cartridge reference label and must be >0.
id	Cartridge information label.
1st trk	First track to be used on the cartridge. If LU2, must be 8 greater than last system track (default=track 0).
#dir trks	Number of directory tracks (1 to 48), (default=1).
#sec/trk	Number of 64-word sectors per track. If LU2/3, parameter is ignored.
bad trks	Bad track list. Up to six track numbers separated by commas.

IN, master security code -- new security code 60

Change master security code. New code is separated from old code by two minus (-) signs.

LI, namr[,format[,ln1[,ln2]]] 10

List contents of a file or lu on list device.

format Specifies list format.

- S Source (default for type 0,3,4 files).
- B Binary (default for all other type files).
- D Directory information only.

ln1 Starting line.

ln2 Ending line.

LL, namr 20

Change current assignment of list device, namr may be either file or lu number.

LO, lu 40

Change lu number of log device where lu is an interactive device.

MC, lu[,P/G[,size[,id[,#dir trks[,label]]]]] 10

Make an unmounted cartridge available for use.

lu Lu number of cartridge to be mounted, it must be in user's session switch table.

P/G Private or group cartridge (session default=P) non-session meaningless, but its space must be provided.

size # of tracks needed on cartridge.

id ASCII identifier of cartridge (default DC00XX; XX is system lu number of terminal).

#dir trks # of tracks used by the file directory (default=1).

label Cartridge reference number to be assigned to the cartridge.

ME[,namr[,clear]]

10

Display contents of user's message file.

namr	File name or non-disc lu to receive messages (default=user's terminal).
clear	1 (clear message file). 0 (do not clear=default).

OF,program

30

Terminate program within caller's current session.

OF,program

60

Terminate any program within the system.

PA[,lu[,message]]

40

Suspend execution of the current job or procedure file, and transfer control to a specified device, and optionally print a message.

lu	Lu to which control transfers (default=log device).
message	1-80 ASCII characters.

PK[,cartridge]

20

Recover tracks and directory entries assigned to purged files and close gaps between files.

cartridge	Cartridge reference number, positive for label or negative for lu (default=all user accessible cartridges).
------------------	---

PU,namr

20

Remove a file and its extents from system.

RN,namr,nuname

20

Change a file name to a new name.

namr	Existing file name and parameters.
nuname	New name unique to the cartridge, namr subparameters may not be changed.

RP,namr,program 30

Restore program file "namr" using the ID segment of "program".

RP,namr 30

Restore program file "namr", which must be a type 6 file on LU2/LU3.

RP,,program 30

Release "program's" ID segment where "program" is a program with its ID segment in memory.

RT,program 30

Release all disc tracks assigned to a dormant program.

RU,program:IH[,parameters] 30

Schedule "program" for immediate execution, inhibit automatic re-naming feature.

RU[IH],program[,parameters] 30

Schedule "program" for immediate execution. IH inhibits passing of command string.

program Name of program to be executed or namr of type 6 file containing program or procedure file to be executed.

parameters 1-5 parameters to be passed to program or 1-9 parameters passed to a procedure file.

SE[,p1[,p2[,...[p9]]]] 40

Set or clear global parameters 1G-9G where p1-p9 are values to be converted to global parameters. If all parameters omitted, globals are nulled. If any one parameter omitted, corresponding global unchanged.

SL[,lu] 10

Display linkage information for session logical unit number.

lu Session logical unit number (default=list information for all session lu's in user's Session Switch Table).

SL,lu[,namr[,attribute[,outlu[,priority[,prog]]]]] 30/50

Spool setup and outspool control.

- lu** The session lu to which a spool file is to be associated. The lu must not be LU2 (system disc), LU3 (auxiliary disc), any lu associated with a disc driver, a spool lu, or if in a job system LU5 (standard spool input device).
- namr** Name of existing file to be used as a spool file (default=system assigns spool pool file).
- attribute** Defines characteristics of spool access. Any 3 attribute codes can be combined, no delimiters necessary.

attribute codes:

- NO = Queue file for immediate outspool.
- RE = Read only.
- WR = Write only.
- BO = Both read and write.
- WN = Write now.
- BU = Buffered.
- PU = Purge.
- SH = Write spool headers.
- ST = Standard file format.

default for attribute codes:

	outlu specified	outlu not specified
namr specified	WRITE,HOLD, SPOOL HEADERS, SAVE	WRITE,HOLD, SPOOL HEADERS, PURGE
namr not specified	WRITE,HOLD, SPOOL HEADERS, SPOOL POOL FILE	BOTH,HOLD, STANDARD FORMAT,SPOOL POOL FILE

- priority** Outspool priority (default=session — 99, batch — priority of job).
- prog** If specified, program "prog" will be scheduled, with wait, by the spool system when spool lu is closed. Note the spool file will not be outspooled, "prog" must properly dispose of the file. Requires capability of 50.
- outlu** Session lu for outspooling.

SL,session lu,system lu

30/50

Map a new session lu to system lu currently in the user's Session Switch Table. Requires capability of 30.

Add a System lu to user's Session Switch Table. Requires capability of 50.

System lu May be specified as — (a dash) to delete lu mappings which have been created during user's session.

SM,user,namr,message

10

Send message and/or file to another user's message file.

user Log on ID of message recipient, (user.group).

namr Name of file or non-disc lu containing data to be sent.

message String entered from sender's terminal.

**SP,namr[,PR
or [,capability]]
,GR**

30

Place a disc resident program and its ID segment in a type 6 file created by this command. Note that namr can not be an lu. First 5 characters of file name must be identical to disc program name. namr subparameters default to:

security	0
cartridge	-2
file type	type 6
file size	size of program
record size	128

**ST,namr1,namr2[,record format[,eof]
[,file #[, #files]]]**

20

Transfer data from an existing file or lu to another file or lu. namr2 created by this command.

namr1	Source of data.
namr2	Destination of data.
record format	Format of data or EOF control (default=namr1 format or ASCII if non-disc device).
	ASCII ASCII records.
	BReloc Binary relocatable records with checksum.
	BNary Binary records without checksum.
	BAbs Binary absolute records with checksum.
	MTape Magnetic tape ASCII records.
	MS Magnetic tape SIO (System Input/Output) records are expected on namr1. Standard records are written on namr2.
	MSBR Magnetic tape SIO binary relocatable records (same as MS+BR).
eof	Eof control.
	IHibit Inhibits EOF on namr2 and leader punching.
	SAve Save embedded EOF's in namr1.
file #	File or subfile on namr1 where transfer starts (default=1).
#files	Number of files to be transferred from namr1 (default=1).

SV,severity[,global #][,IH]

20

Change the system log device severity code to a new number.

severity	0 display all commands and errors (default). 1 display no commands, all errors. 2 display no commands, no errors except those requiring response. A serious error terminates job. 3 display same as 2, except job not terminated. 4 display no commands, no errors, job not terminated.
global #	Optional G global number (1-9) into which current severity code is to be placed.
IH	Optional parameter to inhibit echo of command entry.

SYcommand

1

Execute RTE system command from FMGR.
Preface command by SY (use no delimiter, e.g., SYTI).

TE,message

10

Send message to the operator via the system console.

TR[,xfer[,parameters]]

1

Transfer control to a file or lu, passing parameters as globals.

- xfer

A negative integer that denotes a transfer back that many files, or the name of a file or lu.
- parameters

The parameters to be set into the globals (1G-9G). Skipped parameters are not changed.

WH[,lu[,option]]
or
WH[,option]

10

Schedule WHZAT program.

- lu

The session lu for display.
- option

default

User's session programs.

AL

Display status of all the suspended and scheduled programs.

SM

Similar to AL except state 3 programs without father son relationships are not listed.

PA

Display status of all partitions.

??[error#]

10

Request FMGR error code explanation.

- error#

FMGR error code (default=last error issued).

***COMMENT LINE**

10

BATCH AND SPOOLING COMMANDS

CONTENT	PAGE
AB	C-2
CS	C-2
EOJ	C-2
JOB	C-3
SL	C-3
RUN	C-5
TL	C-5
XE	C-5

AB

30

Terminate batch job.

CS,lu,attribute

30

Modify or change spool options set up by SL command.

- | | |
|-----------|---|
| lu | lu defined at set up. |
| attribute | one of the following: |
| RWind | reset file to first record. |
| PUrge | change SAve flag to PUrge. |
| SAve | change PUrge flag to SAve. |
| PAss | remove HOld option. |
| ENd | write EOF and terminate spool. Spool file placed in outspool queue (default). |
| BUffer | change to buffering. |
| NBUffer | change to no buffering. |
| NPass | change lu and/or priority information, by specifying the 2 additional parameters:
[.outlu[,priority]]
outlu = new lu
priority = new priority |

EOJ[,RP[,RG]]

30

End of spooled job.

- | | |
|----|--|
| RP | Dismount job's private session cartridges.
(Default=leave mounted.) |
| RG | Dismount job's group session cartridges.
(Default=leave mounted.) |

JOB[,name[:hr:min:sec][,user[,priority[,spool
priority][,sp]]]]

30

Initiate job for spooling.

name	Job name.
:hr:min:sec	CPU time limit for job in hours, minutes, seconds.
user	Session user account ID in the form "user.group/ password". If a job is submitted outside of a session when session is installed this parameter must be specified.
priority	Job priority in range from 1-255 (default = 99).
spool priority	Outspool priority (default=priority).
sp	Specify: NO Outspool now,or NS No outspooling.

SL,lu[,namr[,attribute[,outlu[,priority[,prog]]]]]

30/50

Spool setup and outspool control.

lu	The session lu to which a spool file is to be associ- ated. The lu must not be LU2 (system disc), LU3 (auxiliary disc), any lu associated with a disc driver, a spool lu, or if in a job system LU5 (standard spool input device).
-----------	--

BATCH AND SPOOLING

namr name of existing file to be used as a spool file (default=system assigns spool pool file).

attribute defines characteristics of spool access. Any 3 attribute codes can be combined, no delimiters necessary.

attribute codes:

- NO = Queue file for immediate outspool
- RE = Read only
- WR = Write only
- BO = Both read and write
- WN = Write now
- BU = Buffered
- PU = Purge
- SH = Write spool headers
- ST = Standard file format

default for attribute codes:

	outlu specified	outlu not specified
namr specified	WRITE,HOLD, SPOOL HEADERS, SAVE	WRITE,HOLD, SPOOL HEADERS, PURGE
namr not specified	WRITE,HOLD, SPOOL HEADERS, SPOOL POOL FILE	BOTH,HOLD, STANDARD FORMAT, SPOOL POOL FILE

priority Outspool priority (default=session-99, Batch-priority of job).

prog If specified, program "prog" will be scheduled, with wait, by the spool system when spool lu is closed. Note the spool file will not be outspooled, "prog" must properly dispose of the file. Required capability of 50.

outlu Session lu for outspooling.

RUN,JOB,namr [,priority]

30

Run batch job.

- | | |
|-----------------|--|
| namr | File name of file containing single job to be spooled, or logical unit of input device containing jobs to be spooled; (default=session terminal, or logical unit 5 if outside of session). |
| priority | Priority of job (default=99). |

TL:hr:min:sec

30

Set run time limit.

- | | |
|--------------------|--|
| :hr:min:sec | Time limit for execution of any programs with RU command subsequent to TL command. If omitted, job time limit is used. |
|--------------------|--|

XE,namr[,priority]

30

Job input control.

- | | |
|-----------------|--|
| namr | Identifies input device containing a job to be placed in job queue, may be a logical unit or the name of an existing file. |
| priority | Job priority (default=99). |

GASP COMMANDS

CONTENT	PAGE
RU,GASP	D-2
AB	D-2
CJ	D-2
CS	D-3
DJ	D-3
DS	D-4
EX	D-4
KS	D-4
RS	D-4
SD	D-5
SU	D-5
UP	D-5

GASP

RU,GASP[,lu]

Schedule GASP to prompt for command from lu (default=user's terminal).

RU.GASP.command

Schedule GASP, execute command, then terminate.

lu Logical unit of interactive device on which GASP commands are entered. In a session environment **lu** must be specified if it is different from the session logical unit.

command Any GASP operator command.

^AB,job # ,[u.g]

Before a job is processed, it may be removed with the AB command.

job # Number assigned to job by spool system; use DJ to display job numbers.

u.g Aborts all jobs owned by session account (user.-group).

$$^{\wedge}CJ, job \# < \begin{matrix} , priority \\ H \\ R \end{matrix} >$$

Change job priority or status. Only used for a job in I, R, or RH status.

job # Number assigned to job by spool system; use DJ to display job numbers.

priority New job priority; only allowed before job is active.

H Hold job from processing; changes R status to RH, and I to IH.

R	Release job for processing; changes RH status to R.
----------	---

GASP

^DS[AL][,lu[,u.g]]

Display the spool file name, job number, user.group name, outspool priority, spool status, and the logical unit to which the file is being or will be outspooled.

AL	Causes all spools (session and non-session) to be reported.
lu	Outspool logical unit; only files directed to this lu are displayed; if omitted, all files in the outspool queue are displayed. If in session, lu is the session lu, and the lu displayed is the system lu that the session lu maps to.
u.g	Reports only files belonging to the account of u.g. If the '@' character is used for either the user or group, then all users or groups (or both) are reported.

^EX

Terminate GASP.

^KS< ,spoolfile > [,u.g] ,lu

Remove outspool file from the outspool queue.

spoolfile	Name of spool file to be removed.
lu	Logical unit of device to which file is being outspooled. When running under session, lu is the session logical unit number.
u.g	Kills all spool files owned by session account u.g.

^RS,spoolfile[,lu]

Restart active outspool file from the beginning.

spoolfile	Name of active or active-held spool file in outspool queue.
lu	New logical unit to which file is to be outspooled; if omitted, logical unit previously assigned is used for spool output.

^SD<B[ATCH]
,S[POOL]>

Hold all spooled jobs, all spooled output, or both.

- | | |
|-------------|---|
| B | Hold all pending jobs: spool files are not affected. |
| S | Hold all pending spool files; job processing is not affected. |
| none | If both B and S are omitted, then both job processing and outspooling are held. Inspooling by JOB may continue. |

^SU<B[ATCH]
,S[POOL]>

Start up spool system after it has been shut down with SD.

- | | |
|-------------|---|
| B | Jobs held with SD are released; does not restart outspooling. |
| S | Outspools held with SD are released; does not restart job processing. |
| none | Both jobs and outspools held by SD are restarted. |

^UP[,RS]

Up outspool device.

- | | |
|-----------|--|
| RS | Restart active files from the beginning. |
|-----------|--|

ACCOUNT COMMANDS

CONTENT	PAGE
EX	E-2
HE	E-2
LI	E-2
/A	E-2
TR	E-2
/E	E-2

ACCOUNT

ACCOUNT ID FORMAT

USER.GROUP

@."group" — All users in group.

"user".@ — All users named "USER".

@.@ — All users.

EX[IT]

Terminate the account program.

HE[LP][,keyword[,list]]

List valid commands and scheduled HELP utility.

LI[ST],A[CCT][,<list namr>]

List session wide information.

LI[ST],G[ROUP],<group>[,<list namr>]

List one or more group account entries.

LI[ST],U[SER],<user.group>[,<list namr>]

Lists one or more user account entries.

TE[LL],<user.group>[<,namr>][,<MESSAGE>]

Send a message to a single active user or group, or to all active sessions.

/A

Abort current command.

TR [,control[,list<[NO[ECHO]]] [EC[HO]]]]

Invoke a transfer from within a command.

/E

End current phase.

EDITR COMMANDS

CONTENT	PAGE
RU,EDITR	F-2
CONTROL COMMANDS	F-2
DISPLAY COMMANDS	F-3
LINE EDITS	F-3
CHARACTER EDITS	F-3
SEARCH COMMANDS	F-4
EXCHANGE COMMANDS	F-4
TERMINATIONS	F-4

EDITR

RU,EDITR[,lu[,len]]

- lu** LU of interactive input device (default=user's terminal).
- len** Line length in characters (default=150).

EDITR RESPONSE

/source file?
/

POSSIBLE USER RESPONSES

- 0** Start edit with new, empty file.
- :** Abort EDITR immediately.
- namr** File to be copied to EDITR's work area.
- { }** (blank) Current LS area copies to EDITR's work area.

EDITR prompt character "/" (default).

CONTROL COMMANDS

- Xx** Change prompt character to x.
- CNTL/G** Invoke or delete bell.
- Tx** Change tab control character, leave stops.
- Txsl,...sn** Set tab character to x and stops to sl...sn (default=";"7,21).
- Wcoll,col2** Set window (column) boundaries (default=1,150).
- #xxx start#** Add the column identifier (xxx), and line sequence
increment# numbers.
- =n** Set line length to n (default=150).
- K** Kill trailing blanks
- Mnamr** Merge file "namr" after pending line.

DISPLAY COMMANDS

P	Display and/or edit pending line.
Ln,[lu]	List n lines on LU lu (default= pending and next line).
n	Display line n, make it pending line.
/n	Advance pending line n lines.
+n	Advance pending line n lines.
/n,[lu]	Advance to line n displaying changed lines on lu.
+n,[lu]	Advance to line n displaying changed lines on lu.
N	Display pending line number.
ND	Display line number of current line in destination work area.
H	Display number of characters in pending line.
HL	Display header.
^n	Go back n lines in destination work area (default=1).
S	Display approximate number of words in destination file.

LINE EDITS

P	Edit pending line then display it.
C	Edit pending line then advance pending line.
O	Duplicate pending line.
Rtext	Replace pending line with "text".
ltext	Insert "text" before pending line.
{ } text	Insert "text" after pending line.
-n	Delete n lines (default=1).

CHARACTER EDITS

CNTL/R	Replace characters.
CNTL/I	Insert characters.
CNTL/S	Insert characters.
CNTL/C	Cancel characters.
CNTL/T	Truncate characters.

SEARCH COMMANDS**First Field**

- | | |
|--------------------|---|
| Bfind field | Find a line with "find field" from SOF to EOF. |
| Ffind field | Find a line with "find field" from pending line to EOF. |
| Dfind field | Delete lines from pending line to "find field". |
| Jfind field | Jump to "find field" and make it pending line. |

Find Field

- | | |
|----------------|----------------------------------|
| “,” | Find field tabbed. |
| “esc” | Find field of indefinite length. |
| “/” | Find field within window. |
| “CNTL@” | Find 0 length line. |

EXCHANGE COMMANDS

- | | |
|-----------------------|---|
| Gold/new | Character replace on pending line. |
| Yold/new | Exchange on pending line, display next occurrence of pattern. |
| Xold/new range | Enable exchange pattern over range of lines, with list. |
| Vold/new range | Unconditional character replace, with list. |
| Uold/new range | Unconditional character replace, no list. |

TERMINATIONS

- | | |
|---------------|---|
| A | Abort, leaving source file unchanged. |
| ECnamr | Create a FMGR file with edited version. |
| ER | Replace old file with edited version. |
| ERnamr | Replace existing file "namr" with edited version. |

INTERACTIVE UTILITIES

CONTENT	PAGE
Assembler	G-3
CLOAD	G-3
COMPL	G-3
FORTTRAN	G-2
LOADR Commands	G-5
LOADR Operation	G-4
READT/WRITT	G-6

UTILITIES

FORTRAN AND ASSEMBLER

ASMB
RU, namr1[,namr2[,namr3[,lc[,cs]]]]
FTN4

namr1	Disc file or lu for source file.
namr2	Disc file, lu, or "-" for list. "-" creates file 'namr1 for listing if namr1 begins with &.(default= user's terminal).
namr3	Name of file or "-" for relocatable code. "-" creates file %namr1 for relocatable code if namr1 begins with &.(no default).
lc	Line count per page.
cs	Optional control statement which overrides the source file control statement. Options are as follows:

FORTRAN

- L Output source to list, namr2.
- A Output Assembly listing to namr2.
- T Output symbol table for each main or subprogram to list, namr2.
- M Output a mixed listing of both the source and the object program to list, namr2.
- C Output a cross reference symbol table listing to namr2.
- F Perform page eject.
- D Compile debug lines.
- n Error routine n supplied. n is a decimal digit 1-9 which specifies an error routine, ERRn.
- Q Include the approximate relocatable address of each statement on the listing.

ASSEMBLER

- A Absolute assembly, the addresses generated by the assembler are interpreted as absolute locations in memory.
- R Relocatable assembly, the object program may be loaded anywhere in memory.
- L Output source listing to namr2. This includes both the opcode, and the address of the operand if it is a memory reference instruction.
- Q Output source listing to namr2. This includes only the operand address for single word memory reference instructions, otherwise the entire object code will be listed.
- T Output symbol table to list namr2.
- N,Z Selective assembly, sections of the program are to be included or excluded at assembly time depending upon the option specified.
- C Output a cross reference symbol table to namr2.
- F The floating point machine instructions are to be used instead of the software simulation routines for:
FIX,FLT,FDV,FMP,FAD,FSB.
- X No EAU hardware on machine.

COMPL AND CLOAD**COMPL**

RU, ,namr1[,namr2[,namr3[,cs]]]

CLOAD

These utilities automatically invoke the appropriate compiler or assembler for a specified source file. CLOAD, in addition, schedules LOADR.

namr1	Name of source file.
namr2	Disc file, lu, or "-" for list file. "-" creates file 'namr1 for list file if namr1 begins with &. For CLOAD namr2 must be an lu. (default= user's terminal).
namr3	Name of file or "-" for relocatable code. "-" creates file %namr1 for relocatable code if namr1 begins with &. (no default).
cs	Optional control statement which overrides the source file control statement.

LOADR OPERATION

RU,LOADR[,command[,input[,list[,opcode
[,format[,partn[,size]]]]]]]

command	A command file namr, or input device lu. (default= user's terminal or LU5 if batch).
input	The file name of the relocatable main program or the lu of the relocatable input. (no default).
list	List lu, or file name namr. If a file name is specified, the file must not already exist unless its' name begins with ('). (default= user's terminal or LU5 if batch).
opcode	Default = BGNCTE BG Background program RT Real time program LB Large background program SC System COMMON RC Reverse COMMON NC No COMMON SS Use subsystem global (SSGA). PE Permanent program. TE Temporary program. RP Replace permanent program (do not also specify PE).
format	DB Append DBUGR subroutine to the program. LE List entry points and base page links. NL No listing desired. DC Don't copy, multiple copies of the program are not desired. MP Use current page links, except for external references. CP Use current page links, including external references. BP Use base page links only. (default).
partn	The specific partition number in which program is to be executed.
size	Allows a logical address space larger than the program size. Permits use of a dynamic buffer at the end of the program.

LOADR COMMANDS

SE	Searches the system disc library for undefined externals.
SE,namr	Searches the file namr for undefined externals.
MS,namr	Searches the file namr for undefined externals. The file is searched multiple times to satisfy backward references.
RE,namr	Loads file namr, which may be a program, sub-routine, or segment.
LO,XXXXXB	Changes the load address of the next module to be relocated to the specified address.
LI,YYYY	Set up file YYYY as a library file. Up to 10 files may be specified.
SL	Search all files specified in the library command.
TR,namr	Go to file namr for succeeding LOADR commands.
TR	Return to command file suspended when the undefined external was encountered.
FO	Force load a program or segment.
DI	Print list of undefined externals.
EC	Echo input commands on list device.*
EN	\
EX	
/E	/
AB	Abort the LOADR immediately.
/A	
AS,XX	Assigns the relocated program to partition XX.*
SZ,YY	Allows a logical address space larger than the program size. Permits the use of a dynamic buffer at the end of the program.*
LL,namr	Lu or file name for listing. If a file it must not already exist, unless its name begins with (').*
OP,opcode	Specifies an opcode parameter. See opcode section of LOADR OPERATION.*
FM,format	Specifies a format parameter, see format section of LOADR OPERATION.*

***FOOTNOTE:** Specification of the * commands must precede specification of any RELOCATE, or SEARCH command.

UTILITIES

SAVE DISC CARTRIDGE (WRITT)

RU,WRITT $\left[\begin{array}{l} , -lu(c) \quad [, lu(m)] \\ , +crn \end{array} \right]$

- lu(c) is the logical unit (LU) number of the cartridge to be saved on mag tape.
- +crn is the cartridge reference number (CRN) of the cartridge to be saved on mag tape.
- lu(m) is the logical unit (LU) number of the mag tape unit (default is LU 8). Either a positive or negative LU can be specified.

RESTORE DISC CARTRIDGE (READT)

RU,READT $\left[\begin{array}{l} , -lu(c) \quad [, lu(m) \quad [, P \quad [, size]] \\ , +crn \quad [, G \end{array} \right]$

- lu(c) is the logical unit (LU) number of the cartridge to which the previously saved cartridge is to be restored.
- +crn is the cartridge reference number (CRN) of the cartridge being restored.
- lu(m) is the logical unit (LU) number of the mag tape unit (default is LU 8). Either a positive or negative LU can be specified.
- P designates that the cartridge is to be restored as a private cartridge.
- G designates that the cartridge is to be restored as a group cartridge.
- size is the desired size of the cartridge to which the mag tape contents is to be restored. The size is specified in number of tracks (default is the size of the cartridge saved on the mag tape).

EXEC CALLS

CONTENT	PAGE
I/O, READ/WRITE	H-3
I/O, CLASS GET	H-4
I/O CONTROL	H-5
PROGRAM COMPLETION	H-7
PROGRAM SUSPEND	H-7
PROGRAM SWAP CONTROL	H-8
PROGRAM SCHEDULE	H-8
STRING PASSAGE	H-9
STATUS DEVICE	H-9
STATUS PARTITION	H-10
MEMORY SIZE	H-11
TIME REQUEST	H-11
TIMED EXECUTION (ABSOLUTE)	H-12
TIMED EXECUTION (OFFSET)	H-12
TRACK ALLOCATION	H-13
TRACK RELEASE	H-13
LU LOCK	H-14
RESOURCE MANAGEMENT	H-15

EXEC CODE	PAGE
1	H-3
2	H-3
3	H-5
4	H-13
5	H-16
6	H-7
7	H-7
8	H-8
9	H-8
10	H-8
11	H-11
12	H-12
13	H-9
14	H-9
15	H-13
16	H-13
17	H-3
18	H-3
19	H-5
20	H-3
21	H-4
22	H-8
23	H-8
24	H-8
25	H-10
26	H-11

PARAMETERS

Parameters enclosed in [square] brackets are optional.

Parameters enclosed in <angle> brackets are optional in some cases and required in others.

Single underlined parameters have values returned by the system.

Double underlined parameters have values returned by the system in some cases, and user supplied in other cases.

I/O, READ/WRITE

EXEC

1,2,17,18,20

CALL EXEC (ICODE, ICNWD, IBFR, ILEN
[, IPRM1][IPRM2], ICLAS)

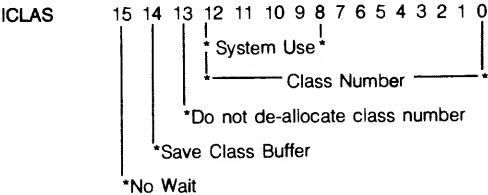
ICODE	1 = READ 2 = WRITE 17 = Class READ 18 = Class WRITE 20 = Class WRITE/READ
ICNWD	Control word, see I/O Control for format. If Z bit (12) is set, an additional control buffer specified by IPRM1, IPRM2 is passed to the driver or to the program doing the GET call.
IBFR	Data buffer.
ILEN	Data length (+ words, - chars).
IPRM1	Optional, or disc track number (for disc transfers), or address of additional control buffer (if Z bit is set).
IPRM2	Optional, or disc sector (for disc transfers), or length of additional control buffer (if Z bit is set).
ICLAS	Class number — required with Class I/O only. ICLAS=0 to allocate a class number.

Returns

Normal I/O	A = Status, EQT wd. 5 (if unbuffered device). B = Transmission log (if unbuffered device).
Class I/O	A = 0 — Request completed. A = -1 — No class number (if no wait bit is set). A = -2 — No memory or buffer limit exceeded (if no wait bit is set). B = Meaningless.

I/O, CLASS GETEXEC21

CALL EXEC (21,ICLAS,IBUFR,ILEN[,IP1][,IP2][,IP3])



- IBUFR Data buffer.
- ILEN Buffer length (+ words, - characters).
- IP1 IPRM1 value returned from a class READ/WRITE or CONTROL call.
- IP2 IPRM2 value returned from a class READ/WRITE or CONTROL call.
- IP3 Returned value of original request code (ICODE).
 - 1 = 17/20 (READ,WRITE/READ)
 - 2 = 18 (WRITE)
 - 3 = 19 CONTROL)

Returns

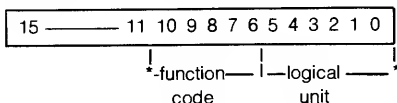
- A-register If data, then A15 = 0 and A = status (EQT wd. 5).
If no data, and no wait bit is set, then A15 = 1 and A= -(numb + 1) where numb is number of requests made to class but not yet serviced by driver.
- B-register If data, then B = transmission log (positive words or characters depending on original request). If no data, then B = meaningless.

I/O CONTROLEXEC
3,19

CALL EXEC(ICODE,ICNWD<,IPRAM>
,ICLAS[,IOP1][,IOP2])

ICODE 3 = Control
 19 = Class Control

ICNWD Control word, see Function Codes below for octal
 bits 6-10.



IPRAM Optional or required for some control functions.

TTY
n space n lines
0 no line feed

LINE PRINTER
+n space n lines
-n top-of-form
0 no line feed

ICLAS Class number — required with class control only.
ICLAS=0 to allocate a class number.

IOP1 (when ICODE = 19) Passed through to Class I/O
IOP2 GET request.

Returns

Normal I/O A = Status, EQT wd. 5 (if unbuffered device).
 B = Meaningless

Class I/O A = Class number
 B = Meaningless

EXEC CALLS

Function Code ICNWD Octal-bits 6-10. See particular driver manual for more information.

- 00 Clear device
- 01 Write end-of-file (MT,CTU)
- 02 Backspace one record (MT,CTU)
- 03 Forward space one record (MT,CTU)
- 04 Rewind (MT,CTU)
- 05 Rewind standby (MT,REWIND CTU)
- 06 Actual status of device (MT,CTU)
- 07 Set end-of-paper tape
- 10 Generate paper tape leader.
- 11 List output line spacing, use IPRAM
- 12 Write gap in case of error (MT)
- 13 Forward space one file (MT,CTU)
- 14 Backward space one file (MT,CTU)
- 15 Conditional top-of-form (LP)
- 20 Enable terminal (CRT)
- 21 Disable terminal (CRT)
- 22 Set time-out, use IPRAM (CRT)
- 23 Ignore further requests until:
 - a) Device queue empty
 - b) Input request encountered
 - c) Restore Control request received
- 24 Restore output processing
- 26 Write end-of-data (CTU)
- 27 Locate file number, use IPRAM (CTU)

PROGRAM COMPLETIONEXEC
6

CALL EXEC (6 [,INAME][,INUMB][,IPRM1,...,IPRM5])

CALL RMPAR(IPRM1,...IPRM5) parameter pick-up.

INAME	Terminate INAME or if 0, terminate calling program.
INUMB	0 Normal completion (default). - 1 Serial reusability. 1 Terminate saving resources . 2 Terminate on next schedule: save tracks. 3 Terminate immediately and release tracks.

IPRM1- IPRM5	Up to 5 optional parameters passed to caller next time he executes (INAME = 0 only).
-------------------------	--

Returns

A-register	Unchanged.
B-register	Unchanged or address of optional parameters (if specified).

PROGRAM SUSPENDEXEC
7

CALL EXEC (7)

If program is rescheduled with a GO command that includes parameters, use RMPAR for parameter pick up.

A-register	Unchanged.
B-register	Unchanged or parameter address.

PROGRAM SWAP CONTROLEXEC
22**CALL EXEC (22,IOPTN)**

IOPTN	0 Swap; 1 Do not swap.
-------	---------------------------

Returns

A-register	Meaningless
B-register	Unchanged

PROGRAM SCHEDULEEXEC
8,9,10,23,24**CALL EXEC (ICODE,INAME[,IPRM1,
...,IPRM5][,IBUFR,ILEN])**

ICODE	8 = Segment load 9 = Immediate, wait 10 = Immediate, no wait 23 = Queue, wait 24 = Queue, no wait
INAME	Name of program or segment to be scheduled.
IPRM1- IPRM5	Up to 5 optional parameters passed to program specified in INAME.
IBUFR	Buffer to pass to son. Not used for EXEC 8.
ILEN	Length of buffer (+ words, - characters). Son recovers buffer using String Passage (ICODE = 14) EXEC call. Not used for EXEC 8.

Returns

A-register	0 if schedule successful. Program status if son not scheduled (immediate schedule only). If EXEC 8, the segment's ID segment address.
B-register	Unchanged, or address of IPRM1-IPRM5 if they were used.

STRING PASSAGEEXEC
14**CALL EXEC (14,IRCOD,IBUFR,ILEN)**

IRCOD	Retrieve/write code: 1 Retrieve buffer or command string. 2 Write buffer to father.
IBUFR	Buffer location.
ILEN	Buffer length (+ words, – characters).

Returns

A-register	0 = successful; 1 = no string found.
B-register	Transmission log.

STATUS, DEVICEEXEC
13**CALL EXEC (13,ICNWD,IST1[,IST2][,IST3])**

ICNWD	Lu of device.
IST1	Returned value of EQT word 5, see Device Status table.
IST2	Returned value of EQT word 4, see EQT table.
IST3	Returned value specifying whether device is “up” or “down”.

Returns Meaningless.

STATUS, PARTITION	EXEC 25
--------------------------	--------------------

CALL EXEC (25,IPART,IPAGE,IPNUM,ISTAT)

- IPART Partition number.
- IPAGE Returned value of starting page number.
- IPNUM Returned value of the number of pages with base page included (–1 returned if illegal partition number).
- ISTAT Return for partition status:

15	14	13	12	11	—	7	—	0
----	----	----	----	----	---	---	---	---

RS RT M S C—0—ID SEG NO.

- RS = 1 if partition reserved
- RT = 1 if partition is real time
- M = 1 if partition is mother
- S = 1 if partition is subpartition
- C = 1 if chain is in effect

Returns

- A-register Meaningless.
- B-register Unchanged.

MEMORY SIZEEXEC
26CALL EXEC (26,IFAW,ILMEM,INPGS[,IMAP])

IFAW	Returned value of first available word address after program.
ILMEM	Returned value, the number of words between end of program and end of program's address space.
MPGS	Returned value, number of pages in partition.
IMAP	Returned value of user map (32 word array).

Returns

A-register	Meaningless.
B-register	unchanged.

TIME REQUESTEXEC
11CALL EXEC (11,ITIME[,IYEAR])

ITIME	Return for time value as follows: ITIME (1) = 10's of milliseconds ITIME (2) = Seconds ITIME (3) = Minutes ITIME (4) = Hours ITIME (5) = Julian day of year
IYEAR	Returned value of year (e.g., 1975) (optional).

Returns

A-register	Meaningless.
B-register	Unchanged.

EXEC CALLS

TIMED EXECUTION

(Absolute Start)

EXEC

12

CALL EXEC (12, INAME, IRESL, IMULT,
IHRS, IMIN, ISEC, IMSEC)

INAME Schedule INAME or if 0, schedule calling program.

IRESL Resolution code, see initial offset EXEC 12.

IMULT Execution multiple (set= 0 means run once).

IHRS	}	Defines absolute start time.
IMIN		
ISEC		
IMSEC		

Returns

A-register Meaningless.

B-register Unchanged.

TIMED EXECUTION

(Initial Offset)

EXEC

12

CALL EXEC (12, INAME, IRESL, IMULT, IOFST)

INAME Schedule INAME or if 0, schedule calling program.

IRESL Resolution code.

1 = 10's/ms

2 = Seconds

3 = Minutes

4 = Hours

IMULT Execution multiple (set = 0 means run once).

IOFST Relative start time (negative value) from current time.

Returns

A-register. Meaningless.

B-register Unchanged.

TRACK ALLOCATIONEXEC
4,15**CALL EXEC (ICODE,ITRAK,ISTRK,IDISC,ISECT)**

ICODE 4 = local.
 15 = global.

ITRAK Number of tracks.
 B15 = 1 — Program not suspended if tracks not
 available.
 B15 = 0 — Program suspended if tracks not
 available.

ISTRK Returned value of starting track number (– 1 if tracks
 not available.)

IDISC Returned value of disc lu, where tracks were
 allocated.

ISECT Returned value of number of sectors per track.

Returns Meaningless.

TRACK RELEASEEXEC
5,16**CALL EXEC (ICODE,ITRAK[,ISTRK][,IDISC])**

ICODE 5 = local.
 16 = global.

ITRAK Number of tracks (If ICODE=5, then – 1 = all tracks,
 ISTRK and IDISC unnecessary.)

ISTRK Starting track number.

IDISC Disc lu.

Returns Local.

A-register Meaningless.

B-Register Meaningless.

Returns Global

A-register Status.

 0 = Tracks released.
 – 1 = No tracks released, one in use.
 – 2 = No tracks released, one not global.

B-register Meaningless.

LOGICAL UNIT LOCK PROGRAM CALL

CALL LURQ (IOPTN,LUARY,NOLU)

IOPTN	Octal control word as follows: 0x0000 = Unlock specified lu's. 1x0000 = Unlock all lu's program currently has locked. 0x0001 = Lock with wait specified lu's. 1x0001 = Lock without wait specified lu's. x(bit 14) is no abort bit; 1 = don't abort.
LUARY	Array of lu's to be locked/unlocked. Ignored when IOPTN = 1x0000.
NOLU	Number of lu's to be locked/unlocked. Ignored when IOPTN = 1x0000.

Returns

A-register	0 = Lock successful. -1 = RN not available. 1 = lu already locked.
B-register	Unchanged.

RESOURCE MANAGEMENT

CALL RNRQ (ICODE,IRN,ISTAT)**ICODE**

Control word as follows:

Bits	15	no wait.		
	14	no Abort.		
	13	}	reserved for system use.	
	.			
	.			
	.			
	.			
	5	clear	}	allocate option.
	4	global		
	3	local		
	2	clear	}	set option.
	1	global		
	0	local		

IRN

Resource number.

ISTAT

Status word.

0 = Normal deallocate return.
 1 = RN is clear (unlocked).
 2 = RN is locked locally to caller.
 3 = RN is locked globally.
 4 = No RN available now.
 6 = RN locked locally to other program.
 7 = RN was locked globally when request was made.

Returns**A-register** Meaningless.**B-register** Unchanged.

FMP CALLS

CONTENT	PAGE
APOSN, EAPOS	I-3
CLOSE, ECLOS	I-3
CREAT, ECREA	I-3
CRETS	I-4
FCONT	I-4
FSTAT	I-5
IDCBS	I-6
LOCF, ELOCF	I-6
NAMF	I-6
OPEN, OPENF	I-7
POSTN, EAPOS	I-8
POST	I-8
PURGE	I-8
READF, EREAD	I-9
RWNDF	I-9
WRITE, EWRT	I-9

FMP CALLS

PARAMETERS

Parameters enclosed in [square] brackets are optional.

Parameters enclosed in <angle> brackets are optional in some cases and required in others.

Single underlined parameters have values returned by the system.

Double underlined parameters have values returned by the system in some cases, and user supplied in other cases.

NOTE: The FMP calls beginning with E (eg. ECREA) can define larger files, up to 32767x128 blocks. The FMP calls not beginning with E (eg. CREAT) can only define files up to 16383 blocks, and 32767 records.

IDCB	A 144 word or longer, array used as the data control block (DCB).
IERR	Error return, see FMGR error codes for meaning. If call is successful: OPEN,OPENF IERR= file type. CREAT IERR= number of sectors.
INAM	Six ASCII characters. First character not a blank or number, no embedded blanks, and (+, -:) are not allowed. All six placed must be accounted for, and a Fortran DATA statement can be used to specify INAM.
IBUF	User buffer.
ISC	File security code: <0 read/write protected. =0 not protected (default). >0 write protected only.
ICR	Cartridge reference: >0 cartridge reference number. <0 logical unit number. =0 first one found (default). Order of search; private cartridges, then group cartridges, then system cartridges.
IREC	Next record number, double word for "E" type calls.
IOFF	Block offset of next record.
IRB	Relative block address of next record, double word for "E" type calls.
IDCBS	Actual size of DCB in words (only when IDCB > 144).

APOSN AND EAPOS

APOSN
CALL (IDCB,IERR,IREC<,IRB<,IOFF>>)
EAPOS

Position a disc file (typically type 3) to a known record address. Record addresses are usually obtained through LOCF for APOSN, and ELOCF for EAPOS. IRB and IOFF are required for files with variable length records.

CLOSE AND ECLOS

CLOSE
CALL (IDCB<,IERR>[,ITRUN])
ECLOS

Close DCB and make file available to others, can also truncate file size.

ITRUN One word variable for CLOSE, double word variable for ECLOS.

+n number of blocks to be deleted from the end of the file when it is closed.

-n retain main file, delete extents.

0 standard close (default).

CREAT AND ECREA

CREAT
CALL (IDCB,IERR,INAM,ISIZE,ITYPE
ECREA [,ISC][,ICR][,IDCBS]<,JSIZE>)

Create a disc file.

ISIZE Two entry array describing file size. for CREAT a two word array, for ECREA a double word integer for each entry.

first entry — file size in blocks.

second entry — record length in words (used for type 2 files only).

ITYPE File type (1-32767).

JSIZE Created file size in sectors; optional double word parameter returned by ECREA only.

CRETS

CALL CRETS (IDCB,IERR,NUM,INAM
[,ISIZE][,ITYPE][,ISC]
[,ICR][,IDCBZ][,JSIZE])

CRETS creates a temporary or scratch disc file by making an entry in the File Directory and allocating disc space for the file. CRETS can define files up to 32767x128 blocks in size.

NUM	Scratch file number, a one-word integer 0-99.
ISIZE	A double word integer for each entry. first entry — file size in blocks. second entry — record length in words (used for type 2 files only).
ITYPE	File type (1-32767).
JSIZE	Created file size in sectors; optional double word parameter returned if call was successful.

FCONT

CALL FCONT(IDCB,IERR,ICON1< ,ICON2>)

Control I/O functions on a non-disc type 0 file.

ICON1	Control word, see EXEC 3 call for options.
ICON2	Additional control, see EXEC 3 call for options.

FSTAT

CALL FSTAT(ISTAT[,ILEN][,IFORM][,IOP][,IADD])

Return status of mounted cartridges.

ISTAT Cartridge status buffer returned as FORMAT I or
FORMAT II.

FORMAT I		
WORD	CONTENTS	CARTRIDGE
1	Logical Unit Number	First cartridge
2	Last FMP track	
3	Cartridge Reference Number	
4	Lock Word	
5	Logical Unit Number	Second cartridge
6	Last FMP track	
7	Cartridge Reference Number	
8	Lock Word	
9	Logical Unit Number	
.	.	
.	.	
.	.	
	0 no more discs	

where: Lock word is ID segment address of locking program or 0 (not locked).

FORMAT II		
WORD	CONTENTS	CARTRIDGE
1	Lock word Logical unit #	First cartridge
2	Last FMP track	
3	Cartridge Reference Number	
4	ID	
5	Lock word Logical unit #	Second cartridge
6	Last FMP track	
7	Cartridge Reference Number	
8	ID	
9	Lock word Logical unit #	
.	.	
.	.	
.	.	
	0 no more discs	

where: Lock word is the offset of the ID segment in the Keyword Table or 0 (not locked).

ID identifies who mounted the cartridge.

FMP CALLS

ILEN	Length in words of status buffer (default= 125).
IFORM	Zero for FORMAT I. Non- zero for FORMAT II.
IOP	Type of cartridges to return information about: 1 = all cartridges mounted to the system. 0 = (under session) all private, group, and system cartridges mounted to that session. 0 = (non session) mounted system and non session cartridges.
IADD	0 if entire cartridge list was returned. Non-zero if entire cartridge list could not be returned.

IDCBS

ISIZE=IDCBS(IDCBS)

Return actual DCB buffer area used (use only if IDCBS > 144).

LOCF AND ELOCF

CALL LOCF
 (IDCB,IERR,IREC[],IRB[],IOFF)
 ELOCF [JSEC],JLU[],JTY[],JREC)

Retrieve status and location information from the data control block on an open file.

JSEC	File size in sectors; one word variable for LOCF, double word variable for ELOCF.
JLU	File lu.
JTY	File type.
JREC	Optional return for: record length (type 1 or 2 files). read/write code (type 0 files). meaningless (type 3 and above).

NAMF

CALL NAMF(IDCB,IERR,INAM,MNAM[,ISC][,ICR])

Close the DCB, if open, and rename file INAM to MNAM.

OPEN AND OPENF

```

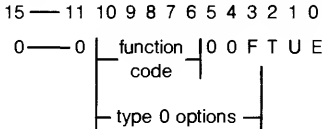
      OPEN
CALL  (IDCB,IERR,INAM
      OPENF  [,IOPTN][,ISC][,ICR][,IDCBS])
  
```

Open a file for access.

INAM ASCII file name, or an integer containing a binary lu (OPENF only).

IOPTN Open control word, defaults are:

- exclusive use, only the calling program can access the file.
- standard sequential output.
- file type defined at creation is used for access.



- E bit 0 exclusive open;
1 non exclusive open.
- U bit 0 non update open;
1 update open.
- T bit 0 file type defined at creation (disc only);
1 force file type to 1.
- F bit 0 use function code defined at creation (type 0 files only);
1 use function code defined in bits 6-10 of IOPTN (for function codes see EXEC 3 call).

FMP CALLS

POSNT AND EPOSN

CALL POSNT (IDCB,IERR,NUR[,IR])
EPOSN

Position files relative to current file position or to a specific record number in any file type.

NUR Record position, a one word variable for POSNT or double word variable for EPOSN.

IR Position mode flag, the relationship between NUR and IR is:

NUR	IR = 0 OR OMITTED RELATIVE POSITION	IR \neq 0 ABSOLUTE POSITION
NUR > 0	Position forward number of records specified	Position to record number specified
NUR = 0	No operation.	No operation
NUR < 0	Position backward number of records specified	Error

POST

CALL POST(IDCB[,IERR])

Write contents of DCB to the disc, and save records in a file opened for non exclusive use. To lock the file for exclusive use with RNRQ call, use the following sequence:

1. call OPEN;
2. read file to pick up resource number;
3. call POST to clear DCB, no data is transferred;
4. call RNRQ to lock the file;
5. call READF to read the record to be modified;
6. modify the record and call WRITF to write it out;
7. call POST to transfer the updated record;
8. call RNRQ to unlock the file.

PURGE

CALL PURGE(IDCB,IERR,INAM<,<ICS><,<ICR>)

Delete named file INAM and all its extents, the file must not be open.

READF AND EREAD

READF
CALL (IDCB,IERR,IBUF[,ILU][,LEN][,NUM])
EREAD

Read a record from an open file to the user buffer. If type 0 file, the number of words should be specified.

ILU	Length of IBUF (read buffer), defaults are: file type = 0 zero length record. file type = 1 128 word record. file type > 1 actual record length.
LEN	Actual read length, set to -1 for EOF.
NUM	A one-word variable (for READF), or double-word variable (for EREAD) used to specify the record number to be read (default= start at current record number).

RWNDF

CALL RWNDF(IDCB[,IERR])

Rewind a magnetic tape or position a disc file to the first record in the file.

WRITEF AND EWRT

WRITEF
CALL (IDCB,IERR,IBUF[,IL][,NUM])
EWRT

Write a record from the user's buffer to an open file. For type 0 or type 3 and above, a specified number of words is written. For type 1 and 2 files the exact record length is written.

IL	Length of write buffer, defaults are: file type = 0 zero length record. file type = 1 128 word record. file type = 2 actual record length. file type > 2 zero length record.
NUM	Record number to be written. (default=start at current record number).

SMP CALLS

CONTENT	PAGE
SPOPEN	J-2
WORKING CALLS	J-3
RETRIEVE RECORD POSITION	J-3
CHANGE RECORD POSITION	J-3

SMP CALLS

PARAMETERS

- ISMP 3 word array containing name of program SMP.
- ISLU Spool lu returned by SPOPN call. Each subsequent
 spool call must specify this lu.

SPOPN

CALL SPOPN(IBUFR,ISLU)

Make a spool file active and ready for use.

- IBUFR 16 word set up buffer structured as follows:

<u>word</u>	<u>contents</u>
0	=0 if no batch input checking desired.
1	>0 session lu for the spool file; or =0 SMP allocates a session lu for the spool file; or =1 a direct map to system lu is set up.
5	security code.
6	cartridge reference number.
7	driver type, in octal.
8	disposition flags:
	<div>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</div> <div><div><div>— — </div><div>— — — — — — — — </div><div>— — — — — — — — </div></div><div>BU BI</div><div>W/R</div><div>ST SP</div><div>HO SA</div></div>
BU	1= buffered; 0= not buffered.
BI	1= batch input; 0 otherwise;
W/R	10B= write; 01B= read; 00B= write/ read.
ST	1= standard file; 0= spool file.
SP	1= spool pool file; 0= user file.
HO	1= hold outspool; 0= outspool now.
SA	1= save file; 0= purge.
9	spool priority (1-9999).
10	spool status (used by SMP,GASP).
11	if batch — job number; if not batch — direc- tory entry number of session program.
12-14	set to 0 or program parameter of SL command.
15	outspool lu.

- ISLU Spool lu return.

WORKING CALLS

CALL EXEC(23,ISMP,XX,ISLU)

- XX
- =1 Change purge to save.
 - =2 Change save to purge.
 - =3 Queue for outspooling.
 - =4 EOF and queue for outspooling.
 - =5 Change spool options; use additional parameters NOL and NPR following ISLU for this call only.
 - NOL new outspool lu (default=previous lu).
 - NPR new outspool priority (default=previous value).
 - =6 Set buffer flag.
 - =7 Clear buffer flag.

RETRIEVE RECORD POSITION

CALL EXEC(23,ISMP,8,ISLU)

CALL RMPAR(IPRM) — for parameter pick up.

- IPRM 5 word array containing pointers to record position.
- | | | |
|--|---|--|
| word 1 = | } | contain an internal coding of the current position of the referenced file. |
| word 2 = | | |
| word 3 = | | |
| word 4 = not used but should be included in array. | | |
| word 5 = not used but should be included in array. | | |

CHANGE RECORD POSITION

CALL EXEC(23,ISMP,9,ISLU,IPRM1,IPRM2,IPRM3)

- IPRM1-3 Record position from the RETRIEVE RECORD call.

TABLES

CONTENT	PAGE
ASCII/BYTES	K-2
ASCII CHARACTERS AND BINARY CODES	K-3
RTE SPECIAL CHARACTERS	K-4
INSTRUCTION CODES IN OCTAL	K-4
BASE SET INSTRUCTION CODES IN BINARY	K-6
EXTENDED INSTRUCTION GROUP CODES	K-8
SYSTEM COMMUNICATION AREA LOCATIONS	K-11
DEVICE REFERENCE TABLE (DRT)	K-15
EQUIPMENT TABLE (EQT)	K-15
DEVICE STATUS TABLE	K-18
EQT WORD 6	K-22
ID SEGMENT	K-23
ID SEGMENT EXTENSIONS	K-26
SESSION CONTROL BLOCK (SCB)	K-27
SYSTEM DISC LAYOUT	K-28
DATA CONTROL BLOCK (DCB)	K-29
CARTRIDGE DIRECTORY FORMAT	K-32
DISC DIRECTORY, CARTRIDGE LABEL ENTRY	K-33
DISC DIRECTORY, FILE ENTRY	K-34
DISC DIRECTORY, TYPE 0 FILE ENTRY	K-35
DISC FILE RECORD FORMATS	K-36
TYPE 6 FILE FORMAT	K-37
RECORD FORMAT: NAM, ENT, EXT, DBL EMA, END, ABSOLUTE	K-38
GLOBAL EQUIVALENCE TABLE	K-45
GENERAL WAIT STATE MESSAGES	K-46
BOOT UP PROCEDURE	K-47

ASCII/BYTES

BYTE POSITION			
CHAR	Left	Right	Dec.
A	040400	000101	65
B	041000	000102	66
C	041400	000103	67
D	042000	000104	68
E	042400	000105	69
F	043000	000106	70
G	043400	000107	71
H	044000	000110	72
I	044400	000111	73
J	045000	000112	74
K	045400	000113	75
L	046000	000114	76
M	046400	000115	77
N	047000	000116	78
O	047400	000117	79
P	050000	000120	80
Q	050400	000121	81
R	051000	000122	82
S	051400	000123	83
T	052000	000124	84
U	052400	000125	85
V	053000	000126	86
W	053400	000127	87
X	054000	000130	88
Y	054400	000131	89
Z	055000	000132	90
a	060400	000141	97
b	061000	000142	98
c	061400	000143	99
d	062000	000144	100
e	062400	000145	101
f	063000	000146	102
g	063400	000147	103
h	064000	000150	104
i	064400	000151	105
j	065000	000152	106
k	065400	000153	107
l	066000	000154	108
m	066400	000155	109
n	067000	000156	110
o	067400	000157	111
p	070000	000160	112
q	070400	000161	113
r	071000	000162	114
s	071400	000163	115
t	072000	000164	116
u	072400	000165	117
v	073000	000166	118
w	073400	000167	119
x	074000	000170	120
y	074400	000171	121
z	075000	000172	122
0	030000	000060	48
1	030400	000061	49
2	031000	000062	50
3	031400	000063	51
4	032000	000064	52
5	032400	000065	53
6	033000	000066	54
7	033400	000067	55
8	034000	000070	56
9	034400	000071	57

BYTE POSITION			
CHAR	Left	Right	Dec.
NUL	000000	000000	0
SOH	000400	000001	1
STX	001000	000002	2
ETX	001400	000003	3
EOT	002000	000004	4
ENO	002400	000005	5
ACK	003000	000006	6
BEL	003400	000007	7
BS	004000	000010	8
HT	004400	000011	9
LF	005000	000012	10
VT	005400	000013	11
FF	006000	000014	12
CR	006400	000015	13
SO	007000	000016	14
SI	007400	000017	15
DLE	010000	000020	16
DC1	010400	000021	17
DC2	011000	000022	18
DC3	011400	000023	19
DC4	012000	000024	20
NAK	012400	000025	21
SYN	013000	000026	22
ET8	013400	000027	23
CAN	014000	000030	24
EM	014400	000031	25
SU8	015000	000032	26
ESC	015400	000033	27
FS	016000	000034	28
GS	016400	000035	29
RS	017000	000036	30
US	017400	000037	31
SPACE	020000	000040	32
!	020400	000041	33
"	021000	000042	34
=	021400	000043	35
\$	022000	000044	36
%	022400	000045	37
&	023000	000046	38
'	023400	000047	39
(024000	000050	40
)	024400	000051	41
*	025000	000052	42
+	025400	000053	43
,	026000	000054	44
-	026400	000055	45
.	027000	000056	46
/	027400	000057	47
:	035000	000072	58
;	035400	000073	59
<	036000	000074	60
=	036400	000075	61
>	037000	000076	62
?	037400	000077	63
@	040000	000100	64
[055400	000133	91
\	056000	000134	92
]	056400	000135	93
^	057000	000136	94
_	057400	000137	95
`	060000	000140	96
~	075400	000173	123
	076000	000174	124
	076400	000175	125
	077000	000176	126
DEL	077400	000177	127

ASCII CHARACTERS AND BINARY CODES

BITS		COLUMN ROW ↓	0 ₀₀	0 ₀₁	0 ₁₀	0 ₁₁	1 ₀₀	1 ₀₁	1 ₁₀	1 ₁₁
b ₇	b ₆ b ₅ b ₄ b ₃ b ₂ b ₁		0	1	2	3	4	5	6	7
0	0 0 0 0	0	NUL	DLE	SP	0	␣	P		p
0	0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0	0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0	0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0	0 1 0 0	4	EOT	OC4	\$	4	O	T	d	t
0	0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0	0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0	0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1	0 0 0 0	8	BS	CAN	(8	H	X	h	x
1	0 0 0 1	9	HT	EM)	9	I	Y	i	y
1	0 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1	0 0 1 1	11	VT	ESC	+	;	K	[k	{
1	0 1 0 0	12	FF	FS	.	<	L	\	l	:
1	0 1 0 1	13	CR	GS	-	=	M]	m	}
1	0 1 1 0	14	SO	RS	.	>	N	^	n	~
1	0 1 1 1	15	SI	US	/	?	O	_	o	OEL

32 CONTROL CODES

64 CHARACTER SET

96 CHARACTER SET

128 CHARACTER SET

EXAMPLE: The representation for the character "K" (column 4, row 11) is.

	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
BINARY	1	0	0	1	0	1	1
OCTAL	1	1		3			

* Depressing the Control key while typing an upper case letter produces the corresponding control code on most terminals. For example, Control-H is a backspace.

RTE SPECIAL CHARACTERS

Mnemonic	Octal Value	Use
SOH (Control A)	1	Backspace (TTY)
EM (Control Y)	31	Backspace (2600)
BS (Control H)	10	Backspace (TTY, 2615, 2640, 2644, 2645)
EOT (Control D)	4	End-of-file (TTY 2615, 2640, 2644, 2645)

INSTRUCTION CODES IN OCTAL

Memory Reference			Ext. Inst. Group
ADA 04(0XX)...	CMA 003000		ADX 105746
ADB 04(1XX)...	CMB 007000		ADY 105756
AND 01(0XX)...	CME 002200		CAX 101741
CPA 05(0XX)...	INA 002004		CAY 101751
CPB 05(1XX)...	INB 006004		CBS 105774
IOR 03(0XX)...	RSS 002001		CBT 105766
ISZ 03(1XX)...	SEZ 002040		CBX 105741
JMP 02(1XX)...	SLA 002010		CBY 105751
JSB 01(1XX)...	SLB 006010		CMW 105776
LDA 06(0XX)...	SSA 002020		CXA 101744
LDB 06(1XX)...	SSB 006020		CXB 105744
STA 07(0XX)...	SZA 002002		CYA 101754
STB 07(1XX)...	SZB 006002		CYB 105754
XOR 02(0XX)...			DSX 105761
↑ Binary			DSY 105771
	Input/Output		ISX 105760
	CLC 1067..		ISY 105770
	CLF 1031..		JLY 105762
	CLD 103101		JPY 105772
	HLT 1020..		LAX 101742
	LIA 1025..		LAY 101752
	LIB 1065..		LBT 105763
	MIA 1024..		LBX 105742
	MIB 1064..		LBY 105752
	OTA 1026..		LDX 105745
	OTB 1066..		LDY 105755
	SFC 1022..		MBT 105765
	SFS 1023..		MVW 105777
	SOC 102201		SAX 101740
	SOS 102301		SAY 101750
	STC 1027..		SBS 105773
	STF 1021..		SBT 105764
	STD 102101		SBX 105740
			SBY 105750
	Extended		SFB 105767
	Arithmetic		STX 105743
	ASL 1000(01X)-		STY 105753
	ASR 1010(01X)-		TBS 105775
	DIV 100400		XAX 101747
	DLD 104200		XAY 101757
	DST 104400		XBX 105747
	LSL 1000(10X)-		XBY 105757
	LSR 1010(10X)-		
	MPY 100200		
	RRL 1001(00X)-		
	RRR 1011(00X)-		
	↑ Binary		
Shift-Rotate			
ALF 001700			
ALR 001400			
ALS 001000			
ARS 001100			
BLF 005700			
BLR 005400			
BLS 005000			
BRS 005100			
CLE 000040			
ELA 001600			
ELB 005600			
ERA 001500			
ERB 005500			
NOP 000000			
RAL 001200			
RAR 001300			
RBL 005200			
RBR 005300			
SLA 000010			
SLB 004010			
Alter-Skip			
CCA 003400			
CCB 007400			
CCE 002300			
CLA 002400			
CLB 006400			
CLE 002100			

INSTRUCTION CODES IN OCTAL (CONTINUED)

Floating Point		Fast FORTRAN		Dynamic Mapping System	
FAD	105000	DBLE	105201	DJP	105732
FDV	105060	DDINT	105217	DJS	105733
FIX	105100	SNGL	105202	JRS	105715
FLT	105120	.BLE	105207	LFA	101727
FMP	105040	.CFER	105231	LFB	105727
FSB	105020	.DFER	105205	MBF	105703
.FIXD	105104	ENTP	105224	MBI	105702
.FLTD	105124	.ENTR	105223	MBW	105704
.TADD	105002	.FLUN	105226	MWF	105706
.TDIV	105062	.GOTO	105221	MWI	105705
.TFTD	105126	.NGL	105214	MWW	105707
.TFTS	105122	.PACK	105230	PAA	101712
.TFXD	105106	.PWR2	105225	PAB	105712
TFXS	105102	\$\$ETP	105227	PBA	101713
.TMPY	105042	.XCOM	105215	PBB	105713
.TSUB	105022	.XFER	105220	RSA	101730
.XADD	105001	.XPAK	105206	RSB	105730
.XDIV	105061	..DCM	105216	RVA	101731
.XFTD	105125	..FCM	105232	RVB	105731
.XFTS	105121	..MAP	105222	SJP	105734
.XFXD	105105	..TCM	105233	SJS	105735
.XFXS	105101			SSM	105714
.XMPY	105041			SYA	101710
.XSUB	105021			SYB	105710
				UJP	105736
Scientific Inst. Set		Double Integer		UJS	105737
ALOG	105322	.DAD	105014	USA	101711
ALOGT	105327	.DCO	105204	USB	105711
ATAN	105323	.DDE	105211	XCA	101726
COS	105324	.DDI	105074	XCB	105726
EXP	105326	DDIR	105134	XLA	101724
SIN	105325	.DDS	105213	XLB	105724
SQRT	105321	.DIN	105210	XMA	101722
TAN	105320	.DIS	105212	XMB	105722
TANH	105330	.DMP	105054	XMM	105720
DPOLY	105331	.DNG	105203	XMS	105721
/CMRT	105332	.DSB	105034	XSA	101725
/ATLG	105333	.DSBR	105114	XS	105725
FPWR	105334				
TPWR	105335				





BASE SET INSTRUCTION CODES IN BINARY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D/I	AND	001	0	Z/C	Memory Address										
D/I	XDR	010	0	Z/C											
D/I	IDR	011	0	Z/C											
D/I	JSB	001	1	Z/C											
D/I	JMP	010	1	Z/C											
D/I	ISZ	011	1	Z/C											
D/I	AD*	100	A/B	Z/C											
D/I	CP*	101	A/B	Z/C											
D/I	LD*	110	A/B	Z/C											
D/I	ST*	111	A/B	Z/C											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SRG	000	A/B	0	D/E	*LS	000			†CLE	D/E	‡SL*	*LS	000	
			A/B	0	D/E	*RS	001				D/E		*RS	001	
			A/B	0	D/E	R*L	010				D/E		R*L	010	
			A/B	0	D/E	R*R	011				D/E		R*R	011	
			A/B	0	D/E	*LR	100				D/E		*LR	100	
			A/B	0	D/E	ER*	101				D/E		ER*	101	
			A/B	0	D/E	EL*	110				D/E		EL*	110	
			A/B	0	D/E	*LF	111				D/E		*LF	111	
			NOP	000			000				000			000	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ASG	000	A/B	1	CL*	01	CLE	01		SEZ	SS*	SL*	IN*	SZ*	RSS
			A/B		CM*	10	CME	10							
			A/B		CC*	11	CCE	11							

BASE SET INSTRUCTION CODES IN BINARY (CONTINUED)


15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	IOG		000		1	H/C	HLT		000	Select Code					
					1	0	STF		001						
					1	1	CLF		001						
					1	0	SFC		010						
					1	0	SFS		011						
				A/B	1	H/C	MI*		100						
				A/B	1	H/C	LI*		101						
				A/B	1	H/C	OT*		110						
				0	1	H/C	STC		111						
				1	1	H/C	CLC		111						
					1	0	STO		001	000				001	
					1	1	CLO		001	000				001	
					1	H/C	SOC		010	000				001	
					1	H/C	SOS		011	000				001	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	EAG		000	MPY**		000		010		000			000		
				DIV**		000		100		000			000		
				DLD**		100		010		000			000		
				DST**		100		100		000			000		
				ASR		001		000		0	1				
				ASL		000		000		0	1				
				LSR		001		000		1	0				
				LSL		000		000		1	0				
				RRR		001		001		0	0				
				RRL		000		001		0	0				
Notes: * A or B, according to bit 11. D/I, A/B, Z/C, D/E, H/C coded: 0/1. **Second word is Memory Address.										†CLE	Only this bit is required				
										‡SL*	Only this bit and bit 11 (A/B as applicable) are required.				

EXTENDED INSTRUCTION GROUP CODES IN BINARY

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAX/SAY/ SBX/SBY	1	0	0	0	A/B	0	1	1	1	1	1	0	X/Y	0	0	0
CAX/CAY/ CBX/CBY	1	0	0	0	A/B	0	1	1	1	1	1	0	X/Y	0	0	1
LAX/LAY/ LBX/LBY	1	0	0	0	A/B	0	1	1	1	1	1	0	X/Y	0	1	0
STX/STY	1	0	0	0	1	0	1	1	1	1	1	0	X/Y	0	1	1
CXA/CYA/ CXB/CYB	1	0	0	0	A/B	0	1	1	1	1	1	0	X/Y	1	0	0
LDX/LDY	1	0	0	0	1	0	1	1	1	1	1	0	X/Y	1	0	1
ADX/ADY	1	0	0	0	1	0	1	1	1	1	1	0	X/Y	1	1	0
XAX/XAY/ XBX/XBY	1	0	0	0	A/B	0	1	1	1	1	1	0	X/Y	1	1	1
ISX/IXY/ DSC/DSY	1	0	0	0	1	0	1	1	1	1	1	1	X/Y	0	0	I/D
JUMP INSTRUCTIONS	1	0	0	0	1	0	1	1	1	1	1	1		0	1	0
	JLY = 0 JPY = 1															
BYTE INSTRUCTIONS	1	0	0	0	1	0	1	1	1	1	1	0				
	LBT = 0 1 1 SBT = 1 0 0 MBT = 1 0 1 CBT = 1 1 0 SFB = 1 1 1															
BIT INSTRUCTIONS	1	0	0	0	1	0	1	1	1	1	1	1				
	SBS = 0 1 1 CBS = 1 0 0 TBS = 1 0 1															
WORD INSTRUCTIONS	1	0	0	0	1	0	1	1	1	1	1	1	1	1		
	CMW = 0 MVW = 1															

EXTENDED INSTRUCTION GROUP CODES IN BINARY (CONTINUED)


MEMORY

EXPANSION	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DJP/DJS	1	0	0	0	1	0	1	1	1	1	0	1	1			

DJP = 0 1 0

DJS = 0 1 1

SYB/USB/PAB
PBB/SSM/JRS

1	0	0	0	1	0	1	1	1	1	0	0	1			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

SYB = 0 0 0

USB = 0 0 1


PAB = 0 1 0

PBB = 0 1 1

SSM = 1 0 0

JRS = 1 0 1

XMA/XLA/XSA/
XCA/LFA

1	0	0	0	0	0	1	1	1	1	0	1	0			
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

XMA = 0 1 0


XLA = 1 0 0

XSA = 1 0 1

XCA = 1 1 0

LFA = 1 1 1

MBI/MBF/MBW/
MWI/MWF/MWW

1	0	0	0	1	0	1	1	1	1	0	0	0			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

MBI = 0 1 0

MBF = 0 1 1


MBW = 1 0 0

MWI = 1 0 1

MWF = 1 1 0

MWW = 1 1 1

SYA/USA/
PAA/PBA

1	0	0	0	0	0	1	1	1	1	0	0	1			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--


SYA = 0 0 0

USA = 0 0 1


PAA = 0 1 0

PBA = 0 1 1


EXTENDED INSTRUCTION GROUP CODES IN BINARY (CONTINUED)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMM/XMS/ XMB/XLB/ XSB/XCB/LFB	1	0	0	0	1	0	1	1	1	1	0	1	0			

XMA = 0 0 0
XMS = 0 0 1
XMB = 0 1 0
XLB = 1 0 0
XSB = 1 0 1
XCB = 1 1 0
LFB = 1 1 1

RSA,RVA	1	0	0	0	0	0	1	1	1	1	0	1	1			
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

RSA = 0 0 0
RVA = 0 0 1

RSB/RVB/SJP/ SJS/UJP/UJS	1	0	0	0	1	0	1	1	1	1	0	1	1			
-----------------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

RSB = 0 0 0
RVB = 0 0 1
SJP = 1 0 0
SJS = 1 0 1
UJP = 1 1 0
UJS = 1 1 1

SYSTEM COMMUNICATIONS AREA LOCATIONS

Octal Location	Contents	Description
SYSTEM TABLE DEFINITION		
01645	XIDEX	Address of current program's ID extension
01646	XMATA	Address of current program's MAT entry
01647	XI	Address of index register save area
01650	EQTA	FWA of Equipment Table
01651	EQT#	Number of EQT entries
01652	DRT	FWA of Device Reference Table, word 1
01653	LUMAX	Number of logical units in DRT
01654	INTBA	FWA of Interrupt Table
01655	INTLG	Number of Interrupt Table Entries
01656	TAT	FWA of Track Assignment Table
01657	KEYWD	FWA of keyword block
I/O MODULE/DRIVER COMMUNICATION		
01660	EQT1	Addresses of first 11 words of current EQT entry (see 01771 for last four words)
01661	EQT2	
01662	EQT3	
01663	EQT4	
01664	EAT5	
01665	EAT6	
01666	EQT7	
01667	EQT8	
01670	EQT9	
01671	EQT10	
01672	EQT11	
01673	CHAN	Current DCPC channel number
01674	TBG	I/O address of time-base card
01675	SYSTY	EQT entry address of system TTY

SYSTEM COMMUNICATIONS AREA LOCATIONS (CONTINUED)

Octal Location	Contents	Description
SYSTEM REQUEST PROCESSOR/EXEC COMMUNICATION		
01676	RQCNT	Number of request parameters -1 Return point address
01677	RQRTN	
01700	RQP1	Addresses of request parameters (set for a maximum of nine parameters)
01701	RQP2	
01702	RQP3	
01703	RQP4	
01704	RQP5	
01705	RQP6	
01706	RQP7	
01707	RQP8	
01710	RQP9	
UTILITY PARAMETERS		
01755	TATLG	Negative length of track assignment table
01756	TATSD	Number of tracks on system disc
01757	SECT2	Number of sectors/track on LU2 (system)
01760	SECT3	Number of sectors/track on LU3 (aux.)
01761	DSCLB	Disc address of library entry points
01762	DSCLN	Number of user available library entry points
01763	DSCUT	Disc address of relocatable disc resident library
01764	SYSLN	Number of system library entry points
01765	LGOTK	LGO: LU#, starting track, number of tracks (same format as ID segment word 28)
01766	LGOC	Current LGO track/sector address (same format as ID segment word 26)

SYSTEM COMMUNICATIONS AREA LOCATIONS (CONTINUED)

Octal Location	Contents	Description
UTILITY PARAMETERS, cont'd.		
01767	SFCUN	LS: LU# and disc address (same format as ID segment word 26) Memory protect ON/OFF flag (0/1)
01770	MPTFL	
01771	EQT12	Address of last four words of current EQT
01772	EQT13	
01773	EQT14	
01774	EQT15	
01775 D	FENCE	Memory protect fence address LWA memory background partition
01777	BGLWA	
D letter indicates the contents of the location are set dynamically by the dispatcher.		
SYSTEM LISTS ADDRESSES		
01711	SKEDD	Schedule list
01713	SUSP2	Wait Suspend list
01714	SUSP3	Available Memory list
01715	SUSP4	Disc Allocation list
01716	SUSP5	Operator Suspend list
PROGRAM ID SEGMENT DEFINITION		
01717	XEQT	ID segment address of current program
01720	XLINK	Linkage
01721	XTEMP	Temporary (five words)
01726	XPRIO	Priority word
01727	XPENT	Primary entry point
01730	XSUSP	Point of suspension
01731	XA	A-register at suspension
01732	XB	B-register at suspension
01733	XEO	E and overflow register suspension

SYSTEM COMMUNICATIONS AREA LOCATIONS (CONTINUED)

Octal Location	Contents	Description
SYSTEM MODULE COMMUNICATION FLAGS		
01734	OPATN	Operator/keyboard attention flag
01735	OPFLG	Operator communication flag
01736	SWAP	RT disc resident swapping flag
01737	DUMMY	I/O address of dummy interface flag
01740	IDSDA	Disc address of first ID segment
01741	IDSDP	Position within disk sector
MEMORY ALLOCATION BASES DEFINITION		
01742	BPA1	FWA user base page link area
01743	BPA2	LWA user base page link area
01744	BPA3	FWA user base page link
01745	LBORG	FWA of resident library area
01746	RTORG	FWA of real-time COMMON
01747	RTCOM	Length of real-time COMMON
01750 D	RTDRA	FWA of real-time partition
01751 D	AVMEM	LWA+1 of real-time partition
01752	BGORG	FWA of background COMMON
01753	BGCOM	Length of background COMMON
01754 D	BGDRA	FWA of background partition

DEVICE REFERENCE TABLE (DRT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SUBCHANNEL NO.						(RESERVED)				EOT ENTRY NUMBER						WORD 1
F	DOWNED I/O REQUEST LIST POINTER															WORD 2

WHERE

F (UP/DOWN FLAG) = 0 IF DEVICE IS UP
 = 1 IF DEVICE IS DOWN

EQUIPMENT TABLE (EQT)

WORD	CONTENTS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	I/O REQUEST LIST POINTER <C>														
2	R	DRIVER INITIATION SECTION ADDRESS <A>														
3	R	DRIVER CONTINUATION/COMPLETION SECTION ADDRESS <A>														
4	O <A>	B 	P <E>	S <E>	T <C>	SUBCHANNEL <C>					I/O SELECT CODE # <A>					
5	AV <F>		EQUIPMENT TYPE CODE <A>							STATUS <E>						
6	CONWD (CURRENT I/O REQUEST WORD) <C>															
7	REQUEST BUFFER ADDRESS <C>															
8	REQUEST BUFFER LENGTH <C>															
9	TEMPORARY STORAGE <O> OR OPTIONAL PARAMETER <C>															
10	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>															
11	TEMPORARY STORAGE FOR DRIVER <D>															
12	TEMPORARY STORAGE FOR DRIVER <O>				OR		EOT EXTENSION SIZE, ANY <A>									
13	TEMPORARY STORAGE FOR DRIVER <D>				OR		EOT EXTENSION STARTING ADDRESS, IF ANY <A>									
14	DEVICE TIME-OUT RESET VALUE 															
15	DEVICE TIME-OUT CLOCK <C>															

LEGEND FOR EQT TABLE

R = reserved for system use.

I/O Request List Pointer = points to list of requests queued up on this EQT entry.

D = 1 if DCPC required.

B = 1 if automatic output buffering used.

P = 1 if driver is to process power fail.

S = 1 if driver is to process time-out.

T = 1 if device timed out (system sets to zero before each I/O request).

Subchan = last subchannel addressed.

I/O Select Code = I/O select code for the I/O controller (lower number if a multi-board interface).

AV = I/O controller availability indicator:

0 = available for use.

1 = disabled (down).

2 = busy (currently in operation).

3 = waiting for an available DCPC channel.

Equipment Type Code = type of device on this controller. When this octal number is linked with "DVy," it identifies the device's software driver routine. Some standard driver numbers are:

00 to 07 = paper tape devices or consoles

00 = teleprinter or keyboard control device

01 = photoreader

02 = paper tape punch

05 = 264x-series terminals

07 = multi-point devices

LEGEND FOR EQT TABLE (CONTINUED)

10 to 17 = unit record devices

10 = plotter

11 = card reader

12 = line printer

15 = mark sense card reader

20 to 37 = magnetic tape/mass storage devices

23 = 9-track magnetic tape

31 = 7900 moving head disc

32 = 7905/06/20 moving head disc

33 = flexible disc drives

36 = writable control store

37 = HPIB

40 to 77 = instruments

STATUS = actual physical status or simulated status at the end of each operation (see Device Status Table).

CONWD = combination of user control word and user request code word in the I/O EXEC call (see EQT wd. 6).

Letters in brackets (<>) indicate the nature of each data item as follows:

<A> = fixed at generation or reconfiguration time; never changes

 = fixed at generation or reconfiguration time; can be changed on-line

<C> = set up or modified at each I/O initialization

<D> = available as temporary storage by driver

<E> = can be set driver

<F> = maintained by system

DEVICE STATUS TABLE A

Device/Status	7	6	5	4	3	2	1	0
Teleprinter(s) Photoreader(s) Punch(s) DVR00	X	—	End of I/O Tape	—	—	STL	TEN	—
263x 264x Terminal	BF	—	CD	—	—	—	TEN	—
Cartridge Tape Unit DVR05, DVA05	EOF	TLP	EOT	RE	LCA	CWP	EOD	CNE/ DB
2892A Card Reader DVR11	HE/ SOR	SF	HE/ SF	PF	TE/ PF	OL	ICC/ HF	RNR
2607 Line Printer	—	TOF	—	ID	PSE	OL	—	—
2610/2614 Line Printer	—	TOF	—	ID	SSE	PO	—	—
2613/17/18 Line Printer	—	TOF	—	ID	ON	NR	V9	V12
2631 Line Printer DVA12	—	TOF	—	BR	ON	PO	—	—
2608A Line Printer DVB12	PW	TOF	S8	VI	ON	NR	V9	V12
2607A Line Printer DVR12	TUF	DM	ON	RY	—	—	APE	—

DEVICE STATUS TABLE A (CONTINUED)

Device/Status	7	6	5	4	3	2	1	0
7261A Card Reader DVR15	EOF	—	HF/ SF	PF	—	—	DE	RNR
7970 Mag Tape DVR23	EOF	ST	EOT	TE	I/O R	NW	PE/ TE	OL
7900 Moving Head Disc DVR31	—	NR	EOT	AE	FC	SC	DE	EE
79XX Disc Drives DVR32	PS	FS	HF	FC	SC	NR	DB	EE
79XXH, 9895 Disc Drives DVA32	PS	FS	HF	FC	SC	NR	DB	EE
See Status Table B DVR33								
59310B HPIB DVR37	—	EF	II/O	NOA	SRQA	IFC	TO	—

See Page K-21 for Key.

DEVICE STATUS TABLE B

DVR33

127323A, 12733A Disc Drives

Bits 0-7	Meaning
00000000	No Error
00000011	No Drive Power
00000101	Door Open
00000111	No Disc
00001011	Record Not Found
00001101	Track Not Found
00001111	Data Checkword Error
00010001	Data Overrun
00010011	Read "Tight Margin" Error
00011111	Transfer Incomplete
00100001	Data Block too long
00100000*	End of Track (Access track > 66)
01000000*	Disc Change
10000000*	Disc Write Protected

DVA47

Serial Link Driver

Bits 0-7	Meaning
00000001	Time out occurred
00000010	Hardware Failure
00000011	Hardware Failure on Controller
00000100	Bad System Configuration
00000101	Illegal Request

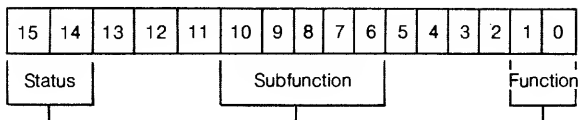
DEVICE STATUS TABLE KEY

AE	= Address Error
AF	= Abort Flag (NR (Bit = 7 = 0) has occurred during since last data transfer)
APE	= Auto Page Eject
BF	= Buffer Flushed
BR	= Buffer Ready
BT	= Broken Tape
CD	= Control-D Entered
CE	= Compare Error
CNI	= Cartridge Not Inserted
CWP	= Cartridge Write Protected
DB	= Device Busy
DE	= Data Error
DM	= Demand (1= idle)
DR	= Disc Ready
EE	= Error Exists
EF	= EQT Extension Area Full
EOD	= End of Data
EOF	= End of File
EOT	= End of Track
FC	= Flagged Track
FS	= Driver Format Switch is Set
HE	= Hopper Empty
HF	= Hardware Fault
ICC	= Illegal Card Code
ID	= Idle
IFC	= IFC Detected
II/O	= Illegal I/O Request
I/OR	= I/O Reject
LCA	= Last Command Aborted
LCF	= Last Character Flag
NE	= No Error
NOA	= Non-existent alarm program
NR	= Not Ready
NW	= No write (ring missing or rewinding)
OL	= Off Line
ON	= On Line
PD	= Pen Down
PE	= Parity Error

DEVICE STATUS TABLE KEY (CONTINUED)

PF	= Pick Fail
PW	= Power Fail
PO	= Paper Out
PS	= Protect Switch Set
PSE	= Print Switch Enabled
RE	= Read Error
RNR	= Reader Not Ready
RX	= Ready (0= Power On)
SAC	= Sector Address Coincidence
SC	= Seek Check
SF	= Stacker Full
SOR	= EOF Switch on during Read
SSE	= Start Switch enabled
ST	= Start of Tape
STL	= Stall required in program
S8	= Set is 8 LPI
TE	= Timing Error
TEN	= Terminal Enabled
TLP	= Tape at Load Pt
TO	= Device Time Out
TOF	= Top of Form
VI	= VFC Initialized
V9	= VFU Chan 12 detected
V12	= V9 VFU Chan 9 detected
WE	= Currently addressed track is write enabled
.X	= Driver internal use

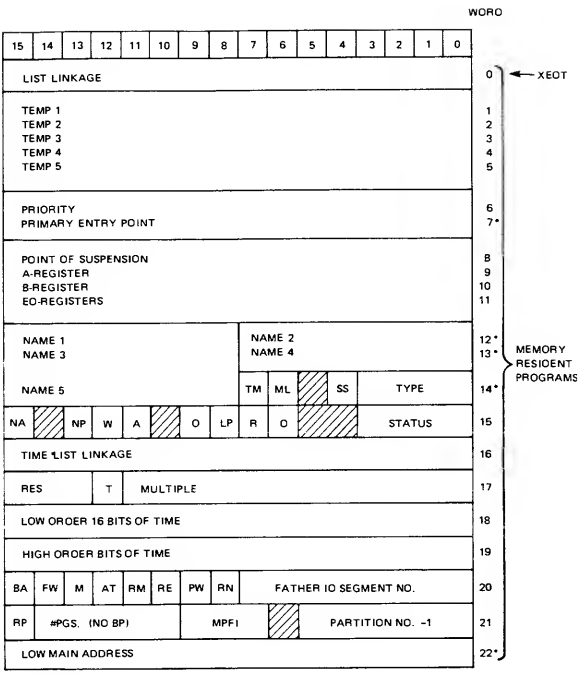
EQT WORD 6





00 — standard call	00000 = clear controller	01 — READ call
10 — buffered call	(if function = 11 =	10 — WRITE call
11 — Class call	CONTROL call)	11 — CONTROL call

Other subfunctions are
driver specific and may or
may not be defined

ID SEGMENT



ID SEGMENT (CONTINUED)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
HIGH MAIN ADDRESS + 1															23 *	
LDW BASE PAGE ADDRESS															24 *	
HIGH BASE PAGE ADDRESS + 1															25 *	
LU	PRDGRAM TRACK								SECTOR						26 *	
LU	SWAP TRACK								ND. TRACKS						27	
ID EXTENSION NO.						EMA SIZE									28	
HIGH ADDRESS + 1 DF LARGEST SEGMENT															29	
TIMESLICE WORD															30	
SEDCNT			DC	CP			SESSION ID								31	
SESSION WORD															32	


MEMDRY
RESIDENTS

* = WORDS USED IN SHORT ID SEGMENTS

ID SEGMENT LEGEND

TM	= temporary load (copy of ID segment is not on the disc)
ML	= memory lock (program may not be swapped)
SS	= short segment (indicates a nine-word segment)
TYPE	= specified program type (1-5)
NA	= no abort (instead, pass abort errors to program)
NP	= no parameters allowed on reschedule
W	= wait bit (waiting for program whose ID segment address is in word 2)
A	= abort on next list entry for this program
O	= operator suspend on next schedule attempt
LP	= load in progress; program is being dispatched from disc.
R	= resource save (save resources when setting dormant)
D	= dormant bit (set dormant on next schedule attempt)
Status	= current program status
T	= time list entry bit (program is in the time list)
BA	= batch (program is running under batch)
FW	= father is waiting (father scheduled with wait)
M	= Multi-Terminal Monitor bit
AT	= attention bit (operator has requested attention)
RM	= reentrant memory must be moved before dispatching program
RE	= reentrant routine now has control
PW	= program wait (some other program wants to schedule this one)
RN	= Resource Number either owned or locked by this program
RP	= reserved partition (only for programs that request it)
DC	= don't copy flag
CP	= copy flag
MPFI	= memory protect fence index

ID SEGMENT EXTENSION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
NS	CURRENT MSEG NO.										# PAGES MSEG					WORD 0
MSEG START PAGE (LOGIC.)					OE	(PHYSICAL) EMA START PAGE										WORD 1
						# TRACKS FOR EMA SWAP										WORD 2

WHERE:

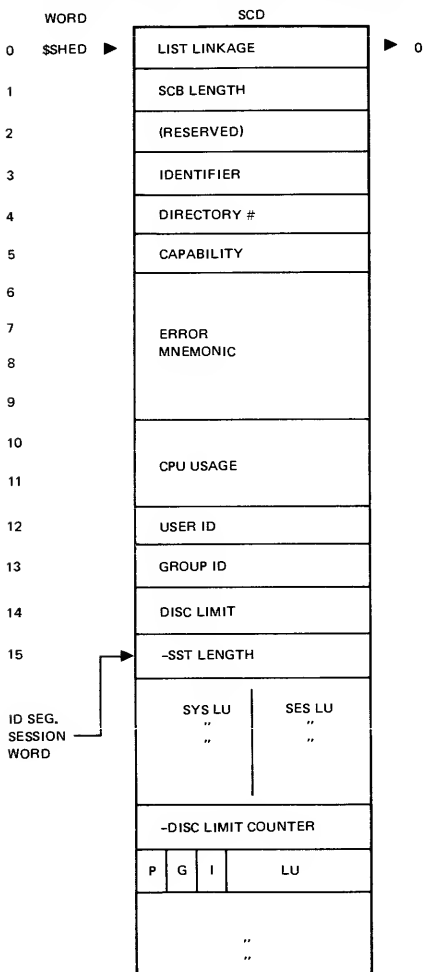
NS = 0 IF THE MSEG IS POINTING TO A STANDARD SEGMENT
OF THE EMA (SET UP BY EMAP)

1 IF THE MSEG IS POINTING TO A NON-STANDARD
SEGMENT (SET UP BY EMIO)

DE = 0 IF THE EMA SIZE WAS SPECIFIED BY THE USER

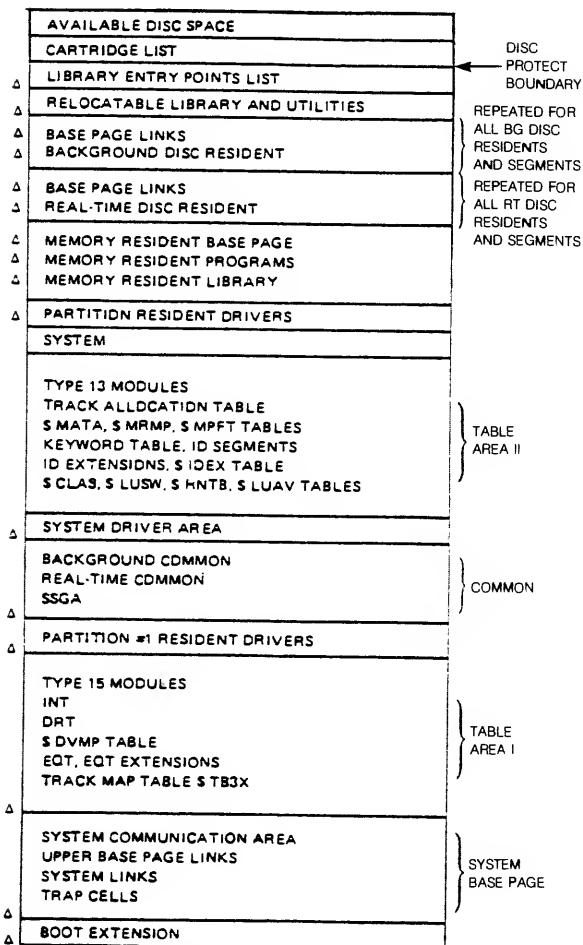
1 IF THE EMA SIZE IS ALLOWED TO DEFAULT TO THE
MAXIMUM SIZE AVAILABLE TO THE SYSTEM.

SESSION CONTROL BLOCK (SCB)



P = ADDED SST ENTRY FOR THIS DISC
 G = THIS IS A GROUP CARTRIDGE
 I = THIS DISC CARTRIDGE IS INACTIVE

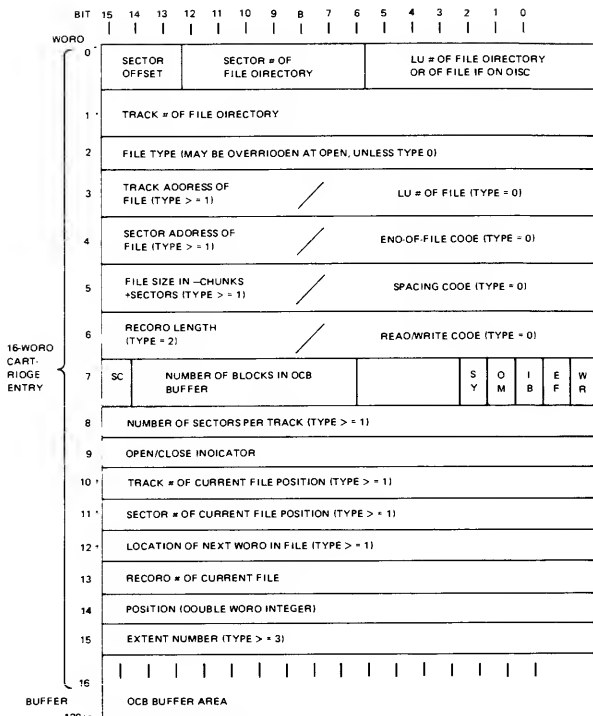
RTE-IVB SYSTEM DISC LAYOUT



Δ SECTOR BOUNDARIES

*INCLUDES ONE SYSTEM-RESERVED TRACK

DATA CONTROL BLOCK



* FILE DIRECTORY ADDRESS

† CURRENT POSITION IN FILE

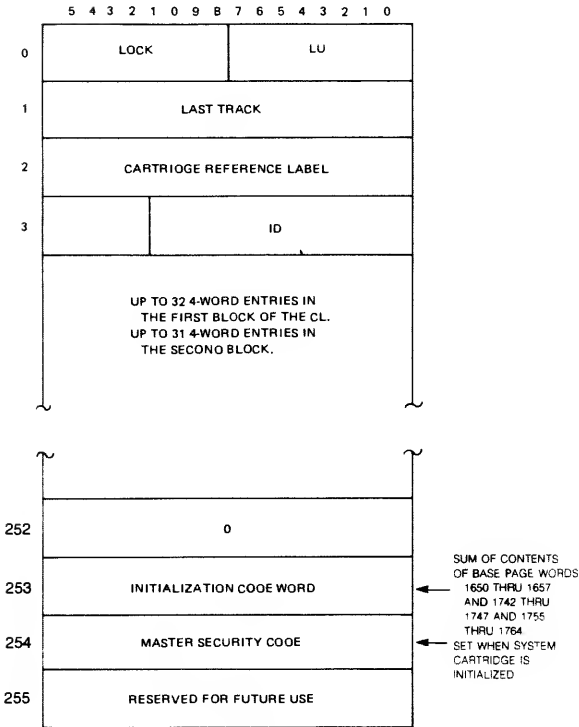
LEGEND FOR DATA CONTROL BLOCK

WORD	CONTENT
4 End-of-File Code, type 0 file:	01 lu = EOF on Magnetic Tape 10 lu = EOF on Paper Tape 11 lu = EOF on Line Printer
5 Spacing Code, type 0 file:	bit 15 = 1 — backspace legal bit 0 = 1 — forward space legal
6 Read/Write Code, type 0 file:	bit 15 = 1 — input legal bit 0 = 1 — output legal
7 Security Code Check/Open Mode/Buffer Size/In Buffer/To Be Written/EOF Read Flag, all file types	
(SC) Security Code Check	bit 15 = 1 — security codes agree = 0 — security codes do not agree
DCB Buffer:	bits 14-7 = Number of blocks in DCB buffer
(SY) System Disc:	bit 4 = 1 file is on a system disc = 0 not on a system disc

LEGEND FOR DATA CONTROL BLOCK (CONTINUED)

WORD	CONTENT
(OM) Open Mode:	bit 3 = 1 — update open 0 — standard open
(IB) In Buffer Flag:	bit 2 = 1 — data in DCB buffer = 0 — data not in DCB buffer
(EF) EOF Read Flag:	bit 0 = 1 — EOF has been read = 0 — EOF has not been read
(WR) To Be Written:	bit 0 = 1 — data in DCB buffer to be written = 0 — data in DCB buffer not to be written
9	Open/Close Indicator: if open, contains ID segment location of program performing open. If closed, set to zero.

CARTRIDGE DIRECTORY FORMAT



LOCK = 0 IF NOT LOCKEO; ELSE IS
KEYWORD TABLE OFFSET OF ID SEGMENT
AODRESS OF LOCKING PROGRAM

LOCKEO DISCS ARE AVAILABLE
ONLY TO THE LOCKER.

ID IDENTIFIES TO WHOM THE CARTRIOGE
IS MOUNTEO.

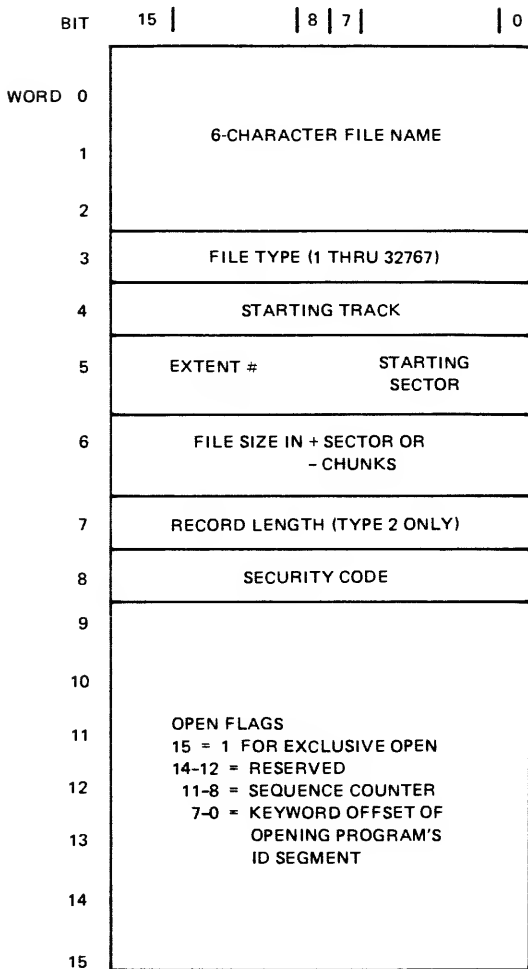
ID = 0000 → NON-SESSION
 ID = 7777 → SYSTEM CARTRIOGE
 0 < ID < 7777 → SESSION MONITOR
 GROUP OR PRIVATE CARTRIDGE

NOTE: WORDS 124, 125, 126, AND 127
 ARE UNIQUE ONLY IN THE SECONO BLOCK
 OF THE CL. THE FIRST BLOCK WILL HOLD
 32 ENTRIES IN WORDS 0 THROUGH 127.

DISC DIRECTORY CARTRIDGE LABEL ENTRY

WORD	CONTENT	IN PARAMETERS
15	0	
0		BIT 15 SET TO DISTINGUISH CARTRIDGE ENTRY FROM FILE ENTRY 6 CHARACTER CARTRIDGE LABEL.
1		
2		
3	CARTRIDGE REFERENCE NUMBER	
4	FIRST AVAILABLE TRACK FOR FMP	
5	RESERVED NEXT AVAILABLE SECTOR	
6	NUMBER OF SECTORS PER TRACK	
7	LOWEST DIRECTORY TRACK (LAST FILE TRACK + 1)	
8	NUMBER OF TRACKS IN DIRECTORY (NEGATIVE VALUE)	
9	NEXT AVAILABLE FMP TRACK	
10	FIRST BAD TRACK	
	• • • • • • • •	
15	SIXTH BAD TRACK	

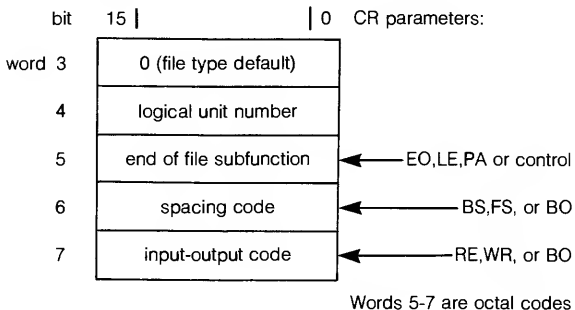
DISC DIRECTORY FILE ENTRY



WORD 0 = 0 IF THE LAST ENTRY IN
 DIRECTORY; = -1 IF FILE IS PURGED

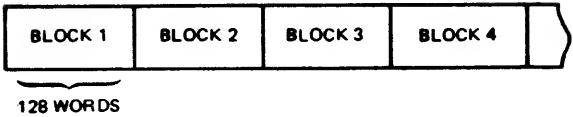
DISC DIRECTORY TYPE 0 FILE ENTRY

The entries for non-disc (type 0) files differ from those for disc files in words 3 through 7:

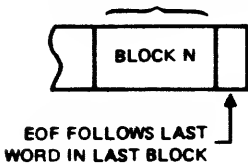


DISC FILE RECORD FORMATS

Fixed Length Formats (Types 1 and 2)



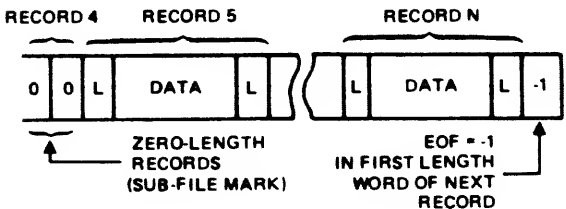
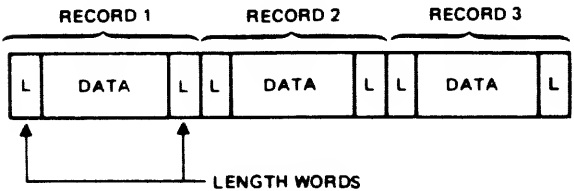
1st 127 WORDS OF LAST BLOCK



Type 1 Record length = Block length = 128 words

Type 2 Record length is user defined; may cross block boundaries but not past EOF

Variable Length Formats (Types 3 and Above)



TYPE 6 FILE FORMAT

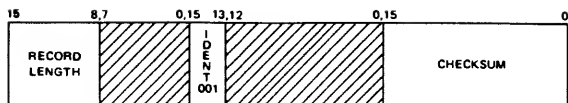
Files created by the SP command as memory-image program files are always accessed as type 1 files (fixed length, 128-words per record).

WORD	CONTENT	
0	-1	EOF UNLESS FORCED TO TYPE 1
1-5	NOT USED	
6	PRIORITY	
7	PRIMARY ENTRY POINT	
8-13	NOT USED	
14	PROGRAM TYPE	
15-16	NOT USED	
17-19	TIME PARAMETERS	
20	SUBSTATUS 1 - WORD 21 OF ID SEGMENT	
21	SUBSTATUS 2 - WORD 22 OF ID SEGMENT	
22	LOW MAIN ADDRESS	
23	HIGH MAIN ADDRESS	
24	LOW BASE-PAGE ADDRESS	
25	HIGH BASE-PAGE ADDRESS	
26-27	NOT USED	
28	ID EXT. #/EMA SIZE	
29	HIGH ADDRESS +1 OF LARGEST SEGMENT	
30-32	NOT USED	
33	CHECKSUM OF WORDS 0 - 32	SUM OF CONTENTS OF WORDS 1650 THRU 1657 AND WORDS 1742 THRU 1747 AND 1755 THRU 1764 IN BASE PAGE
34	SETUP CODE WORD	
35	ID EXTENSION - WORD 1	
36	ID EXTENSION - WORD 2	
37	NOT USED	
38	OWNER ID	IF SIGN BIT SET, PROGRAM FILE PROTECTED TO THIS USER ID
39	OWNER'S GROUP ID	IF SIGN BIT SET, PROGRAM FILE PROTECTED TO THIS GROUP ID
40	CAPABILITY LEVEL REQUIRED	MINIMUM CAPABILITY REQUIRED TO RU OR RP THIS PROGRAM.
41-127	NOT USED	

REMAINDER OF FILE IS AN EXACT COPY OF THE PROGRAM BEING SAVED.

NAM RECORD

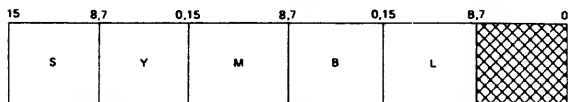
CONTENT



WORD 0

WORD 1

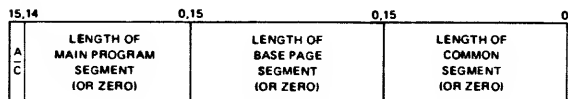
WORD 2



WORD 3

WORD 4

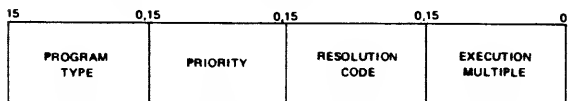
WORD 5



WORD 6

WORD 7

WORD 8

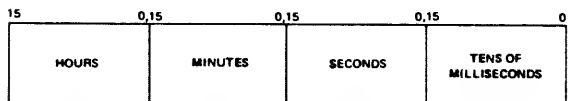


WORD 9

WORD 10

WORD 11

WORD 12



WORD 13

WORD 14

WORD 15

WORD 16



WORD 17

WORD n
($n \leq 60$)

 HATCH-MARKED AREAS SHOULD BE ZERO-FILLED WHEN THE RECORDS ARE GENERATED

 CROSS-HATCHED AREAS SHOULD BE BLANK-FILLED WHEN THE RECORDS ARE GENERATED.

EXPLANATION

RECORD LENGTH = 9-60 WORDS

IDENT = 001

CHECKSUM: ARITHMETIC TOTAL OF ALL WORDS IN RECORD EXCLUDING WORDS 1 AND 3.

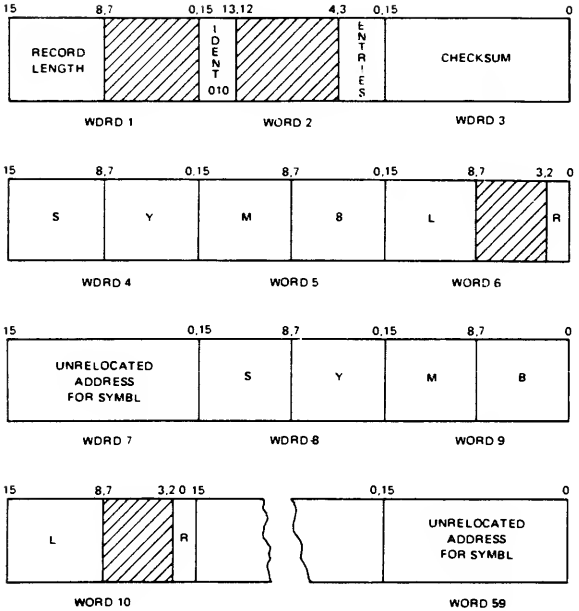
SYMBL. FIVE CHARACTER NAME OF PROGRAM

A/C. BINARY TAPE PRECESSION
= 0 IF ASSEMBLER PRODUCED, OR LENGTH IS EXACT

= 1 IF COMPILER PRODUCED, AND LENGTH IS UNKNOWN

ENT RECORD

CONTENT



EXPLANATION

RECORD LENGTH = 7-59 WORDS

IDENT = 010

ENTRIES 1 TO 14 ENTRIES
PER PROGRAM, EACH ENTRY
IS FOUR WORDS LONG

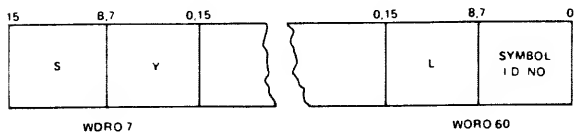
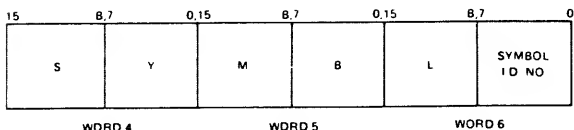
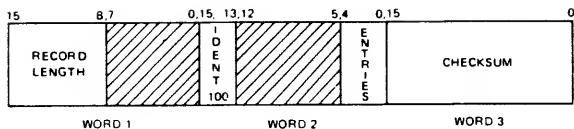
SYMBL 5 CHARACTER ENTRY
POINT SYMBOL

R: RELOCATION INDICATOR
= 0 IF PROGRAM RELOCATABLE
= 1 IF BASE PAGE RELOCATABLE
= 2 IF COMMON RELOCATABLE
= 3 IF ABSOLUTE
= 4 MICROCODE REPLACEMENT

WORDS 4 THROUGH 7 ARE
REPEATED FOR EACH
ENTRY POINT SYMBOL

EXT RECORD

CONTENT



EXPLANATION

RECORD LENGTH = 6-60 WORDS

IDENT = 100

ENTRIES 1 TO 19 PER
RECORD; EACH ENTRY
IS THREE WORDS LONG

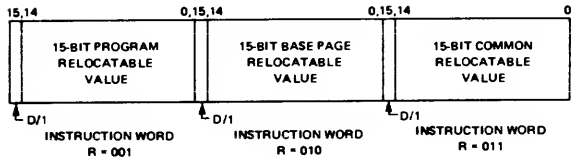
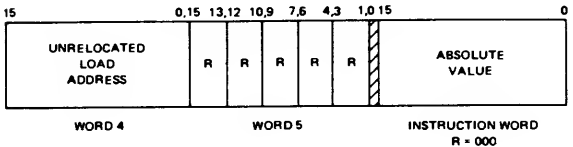
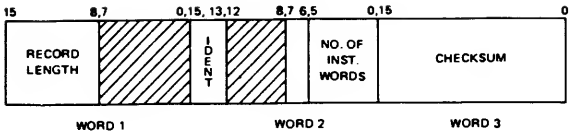
SYMBL 5 CHARACTER
EXTERNAL SYMBOL

SYMBOL ID NO NUMBER
ASSIGNED TO SYMBL FOR
USE IN LOCATING
REFERENCE IN BODY
OF PROGRAM.

WORDS 4 THROUGH 6 REPEATED
FOR EACH EXTERNAL
SYMBOL (MAXIMUM OF
19 PER RECORD)

DBL RECORD

CONTENT



EXPLANATION

RECORD LENGTH = 6-60 WORDS
IDENT = 011

Z/C: RELOCATION OF LOAD ADDRESS

- = 0 FOR BASE PAGE
- = 1 FOR PROGRAM
- = 2 FOR ABSOLUTE
- = 3 FOR COMMON

NO. OF INST. WORDS 1 TO 45
LOADABLE INSTRUCTION WORDS PER RECORD

RELOCATABLE LOAD ADDRESS
STARTING ADDRESS FOR
LOADING THE INSTRUCTIONS
WHICH FOLLOW;

R₁: RELOCATION INDICATORS

000 = ABSOLUTE

001 = 15-BIT PROGRAM
RELOCATABLE

010 = 15-BIT BASE PAGE
RELOCATABLE

011 = 15-BIT COMMON
RELOCATABLE

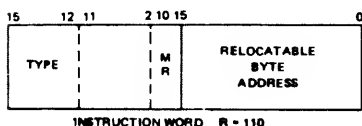
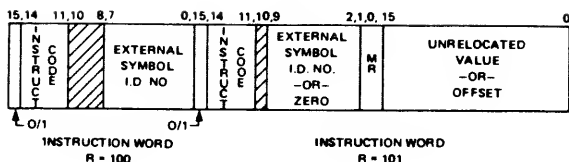
100 = EXTERNAL REFERENCE

101 = MEMORY REFERENCE

R₁ IS RELOCATION INDICATOR
FOR INSTRUCTION WORD₁; R₂,
FOR INSTRUCTION WORD₂; ETC.

DBL RECORD (CONTINUED)

CONTENT



EXPLANATION

D/I: INDIRECT ADDRESSING

0 = DIRECT

1 = INDIRECT

MEMORY REFERENCE INSTRUCTIONS USE TWO WORDS, WITHIN THE TWO-WORD GROUP, "MR" INDICATES RELOCATABILITY OF OPERAND SPECIFIED IN SECOND WORDS:

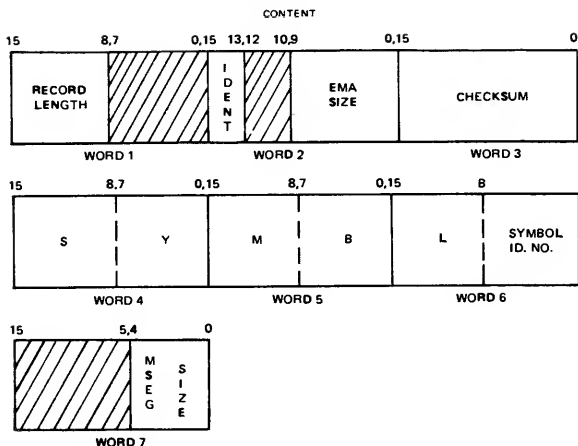
00 = PROGRAM RELOCATABLE

01 = BASE PAGE RELOCATABLE

10 = COMMON RELOCATABLE

11 = ABSOLUTE

EMA RECORD

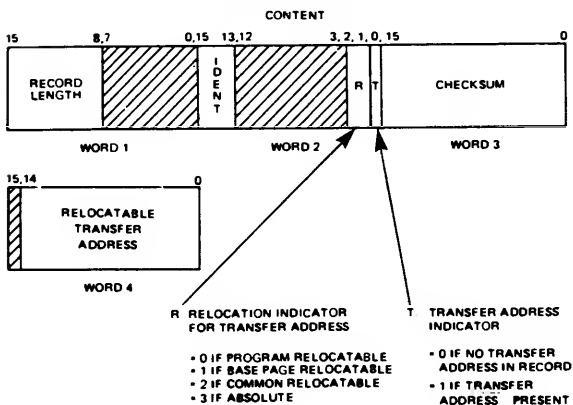


EXPLANATION

RECORD LENGTH = 7 WORD
IDENT = 110

SYMBOL ID. NO.: NUMBER
ASSIGNED TO SYMBL FOR
USE IN LOCATING REFER-
ENCE IN BODY OF PROGRAM.

END RECORD

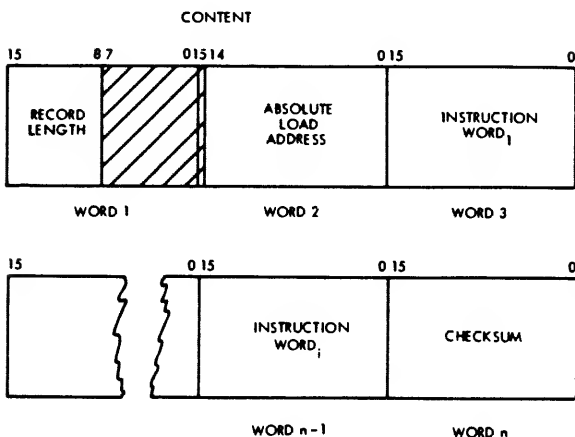


EXPLANATION

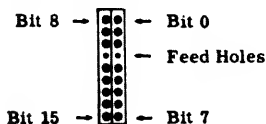
RECORD LENGTH = 4 WORDS
IDENT = 101

ABSOLUTE TAPE FORMAT

Absolute binary code is written to paper tape in the following format:



Each word represents two frames arranged as follows:



EXPLANATION

RECORD LENGTH = NUMBER OF WORDS IN RECORD EXCLUDING WORDS 1 AND 2 AND THE LAST WORD.

ABSOLUTE LOAD ADDRESS:
STARTING ADDRESS FOR
LOADING THE INSTRUCTIONS
WHICH FOLLOW

INSTRUCTION WORDS:
ABSOLUTE INSTRUCTIONS
OR DATA

CHECKSUM: ARITHMETIC
TOTAL OF ALL WORDS
EXCEPT FIRST AND LAST

GLOBAL EQUIVALENCE

S	G	P
0	-2	-48 Type
		-47 1
		-46 2
		-45 3
1	-1	-44 Type
		-43 1
		-42 2
		-41 3
2	0	-40 Type
		-39 1
		-38 2
		-37 3
3	1	-36 Type
		-35 1
		-34 2
		-33 3
4	2	-32 Type
		-31 1
		-30 2
		-29 3
5	3	-28 Type
		-27 1
		-26 2
		-25 3
6	4	-24 Type
		-23 1
		-22 2
		-21 3
7	5	-20 Type
		-19 1
		-18 2
		-17 3
8	6	-16 Type
		-15 1
		-14 2
		-13 3
9	7	-12 Type
		-11 1
		-10 2
		- 9 3
10	8	- 8 Type
		- 7 1
		- 6 2
		- 5 3
11	9	- 4 Type
		- 3 1
		- 2 2
		- 1 3
12	10	0 Type
		1 1
		2 2
		3 3
13	11	4 4
		5 5
		6 6
		7 7
		8 8
		9 9

Last FMGR error

Severity code

Session identifier

User's capability level

The standard values are shown within dark lines.

GENERAL WAIT STATE MESSAGES

(State 3)

MESSAGE	REASON FOR WAIT
LULK lu, LKPRG= progx	The listed program attempted to put a lock on logical unit lu. Program progx already has a lock on lu. The listed program will be rescheduled when progx removes its lock.
RN xx, LKPRG= progx	The listed program attempted to set resource number xx. Program progx already has a lock on the resource number. The listed program will be rescheduled when progx removes the lock.
RESOURCE	The listed program attempted to allocate a resource number. The system has no more resource numbers available. The operating system will reschedule the listed program when a resource number is available.
CLASS #	The listed program requested a class number but the system has no more available. The operating system will reschedule the listed program when a class number becomes available.
CL xx	The listed program is waiting on completion of a class GET to class number xx.
progx	The listed program scheduled progx with wait. The listed program will be rescheduled when progx completes.
progx's QUEUE	The listed program scheduled progx on the queue with wait. progx is not dormant so the listed program must wait. The listed program will be rescheduled after the scheduling of progx completes.
BL,EQT xx	Buffer limit exceeded on the controller in EQT entry xx.
EQLK xxx, LKPRG= PRGA	Program suspended for a locked EQT.
EQLK TABLE FULL	Program attempts to lock an EQT and the EQT table is full.

BOOT UP PROCEDURE

1. Select the S-register for display on the computer front panel.
2. Press CLEAR DISPLAY.
3. Set the S-register bits as follows:

Bits:	Enter:
0-2	Surface number of the disc where the RTE-IVB system subchannel starts.
3-4	0 (reserved).
5	0 for standard boot-up.
6-11	Octal select code of the disc.
12	1 to indicate a manual boot from the S-register.
13	0 (reserved).
14-15	Loader ROM selection (number of the ROM cell containing the Disc Boot Loader).
4. Press STORE.
5. Press PRESET, IBL and PRESET (again) to load contents of Disc Loader ROM.
6. Press RUN.

ERROR CODES

CONTENT	PAGE
ACCOUNT	L-2
ASSEMBLER	L-5
COMPL,CLOAD	L-7
DISC ALLOCATION	L-9
EXEC CALL	L-9
FMGR	L-10
FMGR UNNUMBERED	L-15
FORMAT	L-16
FORTRAN	L-17
GASP	L-22
I/O CALL	L-23
LIBRARY	L-25
LOADR	L-27
LOGON	L-30
LU LOCK	L-30
OUTSPOOL	L-31
READT/WRITT	L-32
RECONFIGURATION	L-34
RESOURCE NUMBER	L-35
SCHEDULE CALL	L-36
SMP	L-37
SYSTEM AND BREAKMODE	L-38
SYSTEM BOOT-UP HALTS	L-39

ACCOUNT ERROR CODES

ACCT-225	Session memory can not be returned to system (reboot)
ACCT-223	Illegal shut down parameter
ACCT-222	Illegal system lu
ACCT-221	Not an active session
ACCT-220	Corrupt station table spares
ACCT-219	Not enough room in file for new table
ACCT-218	Session not shut down
ACCT-215	List NAMR in transfer stack
ACCT-213	Invalid memory request
ACCT-212	Invalid number of SST spares
ACCT-211	Invalid user or group ID not available
ACCT-210	Conflict in SST definition
ACCT-209	Invalid SST entry
ACCT-208	Invalid disc limit
ACCT-207	Invalid capability
ACCT-206	Invalid disc limit
ACCT-205	Invalid command
ACCT-204	Invalid password
ACCT-203	Invalid account name
ACCT-202	Account with this name already exists
ACCT-201	No free accounts
ACCT-099	An Exec request made by D.RTR was aborted.
ACCT-046	Attempt to create extent 256. Make file size of main larger.
ACCT-041	No room in SST

ACCT-040	Lu not found in SST
ACCT-039	Conflict in SST definition
ACCT-035	Already 63 discs mounted to system
ACCT-034	Disc already mounted.
ACCT-033	Not enough room on cartridge
ACCT-032	Cartridge not found
ACCT-030	Value too large for parameter
ACCT-026	Queue full or max pending spools exceeded
ACCT-025	No SPLCON room the SPLCON is full.
ACCT-024	No more batch switches
ACCT-023	No available spool files
ACCT-022	No available spool lu's
ACCT-021	Illegal destination lu
ACCT-020	Illegal access lu
ACCT-019	Illegal access on a system disc
ACCT-018	Illegal lu; lu not assigned to system
ACCT-017	Illegal read/write on Type 0 file
ACCT-016	Illegal Type 0 or file blocks size=0
ACCT-015	Illegal name
ACCT-014	Directory full
ACCT-013	Disc locked
ACCT-012	EOF or SOF error
ACCT-011	DCB not open
ACCT-010	Not enough parameters

ACCT-009	Attempt to use APOSN or force a Type 0 file to Type 1
ACCT-008	File open or lock rejected
ACCT-007	Illegal security code or illegal write on lu2 or 3
ACCT-006	File not found
ACCT-005	Record length illegal
ACCT-004	More than 32767 records in a Type 2 file
ACCT-003	Backspace illegal
ACCT-002	Duplicate file name
ACCT-001	Disc error
ACCT 004	Illegal lu
ACCT 012	Lu not in session switch table
ACCT 013	Transfer stack overflow
ACCT 046	Insufficient capability
ACCT 200	Account not found

ASSEMBLER ERROR CODES

ERROR	PASS	DESCRIPTION
CS	1	Control statement error
DD	1	Doubly defined symbol, a name defined in the symbol table appears more than once.
EN	1	The symbol specified in an ENT statement has already been defined in an EXT statement, or is a label for an EMA pseudo-instruction.
EN UNDEF <symbol>	2	The entry point specified in an ENT statement does not appear in the label field of a machine or BSS instruction. The entry point has been defined in the Operand field of an EXT statement.
IF	1	An IFZ or an IFN follows either an IFZ or an IFN without an intervening XIF. The second pseudo instruction is ignored.
IL	1	Illegal instruction.
	1 or 2	Illegal character, a numeric term used in the Operand field contains an illegal character.
LB	1	Missing label in an EQU, RPL or EMA pseudo-instruction.
M	1 or 2	Illegal operand.

NO	1 or 2 No origin definition, the first statement in the assembly containing a valid opcode following the ASMB control statement is neither an ORG nor a NAM statement.
OP	1 or 2 Illegal Opcode.
OV	1 or 2 Numeric operand overflow, the numeric value of a term or expression has overflowed its limit.
SO	There are more symbols defined in the program than the symbol table can handle.
SY	1 or 2 A label field contains an illegal character or is greater than 5 characters, or a symbolic term in the Operand field is greater than five characters, or the source file contains more than one control statement.
UN	1 or 2 Undefined Symbol.

COMPL AND CLOAD ERROR CODES

- | | |
|--------|--|
| CL- 01 | The input to the COMPL & CLOAD programs must be a source file. |
| CL- 02 | An FMP error was detected on the open request. |
| CL- 03 | An FMP read error occurred. |
| CL- 04 | An FMP error was detected on the close request. |
| CL- 05 | Control statement not in first 10 lines of the program. |
| CL- 06 | The language requested was rejected by the operating system. The language was purged from the system between the 'RP' and the EXEC request. |
| CL- 07 | The language requested in the control statement was recognized but not found. |
| CL- 08 | The language requested exists on the system and COMPL or CLOAD was in the process of 'RP'ing it. When the file was closed an FMP error occurred. |
| CL- 09 | The language requested exists on the system and COMPL or CLOAD was in the process of 'RP'ing it. However, that 'RP' failed because the checksum calculated when the language was 'SP'ed did not match the system checksum. |
| CL- 10 | The language requested exists on the system and COMPL or CLOAD was in the process of 'RP'ing the language. However, during the open request an FMP error occurred. |

- CL- 11 This session has more than 80 spool files currently residing on the spool disc.
- CL- 12 The compiler was aborted.
- CL- 13 The compilation was not successful. Errors or warnings were found.
- CL- 14 This error results when the system is out of ID segments and it is impossible to 'RP' the compiler or LOADR.
- CL- 15 This error means that one of the input parameters was in error.
- CL- 30 CLOAD was trying to 'RP' the LOADR but encountered an FMP error on the close of the file that contained the LOADR.
- CL- 31 CLOAD was trying to 'RP' the LOADR and a checksum error resulted.
- CL- 32 CLOAD was trying to 'RP' the LOADR but encountered an FMP error on the FMP open request.
- CL- 33 If the LOADR was not loaded at generation time or an illegal non supported memory or disc modification has been made.
- CL- 34 The LOADR was loading your program but was aborted abnormally.
- CL- 35 The load was not successful.
- CL- 36 CLOAD was unable to create a copy of the LOADR and even the original LOADR was not available.
- CL- 37 The list device for CLOAD must be an lu because both the compiler and the LOADR must talk to the device.

DISC ALLOCATION ERROR CODES

DR01	Not enough parameters were specified.
DR02	The number of tracks is \leq zero or an illegal logical unit was specified.
DR03	An attempt to release a track assigned to another program was made.

EXEC CALL ERROR CODES

DM	Mapping error. An attempt was made to read/write outside of the mapped address space.
MP	Memory protect error. the call was not an EXEC, \$L1BR, or \$L1BX call.
RE	A re-entrant subroutine attempted to call itself.
RQ	An illegal request code is specified in an EXEC call.
TI	A batch program exceeds the allowed time.

FMGR ERROR CODES

FMGR-102	Illegal D.RTR call sequence
FMGR-101	Illegal parameter in D.RTR call
FMGR-099	Directory manager EXEC request was aborted
FMGR-048	Spool not initialized or SMP cannot be scheduled
FMGR-047	No session lu available for spool file
FMGR-046	Greater than 255 extents
FMGR-041	No room in SST
FMGR-040	Lu not found in SST
FMGR-039	Spool lu not mapped to the spool driver
FMGR-038	Illegal scratch file number
FMGR-036	Lock error on device
FMGR-035	Already 63 discs mounted to system
FMGR-034	Disc already mounted.
FMGR-033	Not enough room on cartridge
FMGR-032	Cartridge not found
FMGR-030	Value too large for parameter
FMGR-026	Queue full or max pending spools exceeded
FMGR-025	No SPLCON room
FMGR-024	No more batch switches
FMGR-023	No available spool files
FMGR-022	No available spool lu's
FMGR-021	Illegal destination lu
FMGR-020	Illegal access lu
FMGR-019	Illegal access on a system disc

FMGR-018	Illegal lu
FMGR-017	Illegal read/write on Type 0 file
FMGR-016	Illegal Type 0 or size=0
FMGR-015	Illegal name
FMGR-014	Directory full
FMGR-013	Disc locked
FMGR-012	EOF or SOF error
FMGR-011	DCB not open
FMGR-010	Not enough parameters
FMGR-009	Attempt to use APOSN or force to 1 a Type 0 file
FMGR-008	File open or lock rejected
FMGR-007	Illegal security code or illegal write on lu2 or 3
FMGR-006	File not found
FMGR-005	Record length illegal
FMGR-004	Record size of Type 2 file is 0 or undefined
FMGR-003	Backspace illegal
FMGR-002	Duplicate file name
FMGR-001	Disc error, the disc is down.
FMGR 000	Break, informative message only no error has occurred.
FMGR 001	Disc error — lu reported, disc associated with the lu is down.
FMGR 002	Initialize lu 2!
FMGR 003	Initialize lu 3!
FMGR 004	Illegal response to FMGR 002 or FMGR 003

FMGR 005	Required track not available — relative TAT position reported
FMGR 006	FMGR suspended
FMGR 007	Checksum error
FMGR 008	D.RTR not loaded
FMGR 009	ID segment not found
FMGR 010	Input error
FMGR 011	Do 'OF,XXXXX,8' on named programs
FMGR 012	Duplicate disc label or lu
FMGR 013	TR stack overflow
FMGR 014	Required ID segment not found
FMGR 015	LS track report
FMGR 016	File must be and is not on lu 2 or lu 3
FMGR 017	ID segment not set up by RP
FMGR 018	Program not dormant
FMGR 019	File not set up by SP on current system
FMGR 020	Illegal Type 0 file
FMGR 021	Illegal disc specified
FMGR 022	Copy terminated
FMGR 023	Duplicate program name
FMGR 041	Program cannot be a segment
FMGR 042	Lu cannot be switched
FMGR 043	Lu not found in SST
FMGR 044	No messages waiting
FMGR 045	Session command only
FMGR 046	Insufficient capability
FMGR 047	Spool set up failed
FMGR 048	Global set out of range

- FMGR 049 Can't run RP'ed program
- FMGR 050 Not enough parameters
- FMGR 051 Illegal master security code
- FMGR 052 Illegal lu
- FMGR 053 Illegal label or ilabel
- FMGR 054 Disc not mounted
- FMGR 055 Missing parameter
- FMGR 056 Bad parameter
- FMGR 057 Bad track not in file area
- FMGR 058 LG area empty
- FMGR 059 Reported track unavailable
- FMGR 060 A re-initialization attempt will raise the first track or lower the directory tracks into the file area and destroy a file.
Enter '??' or 'NO' to stop the reinitialization
Enter 'YES' to continue.
- FMGR 061 Do a "DC" and a "MC" on this CR
- FMGR 062 More than 63 discs
- FMGR 063 Exceeding session disc limit
- FMGR 064 No discs available from disc pool that are big enough.
- FMGR 065 Conflict in SST definition
- FMGR 066 No room in SST
- FMGR 067 Program not found
- FMGR 068 Lu not in variable part of SST
- FMGR 069 Job LOGON failed
- FMGR 070 Sectors/track value too large

- FMGR 071 Do "EX,SP" to save or "EX,RP" to release private cartridges
- FMGR 072 Lu not interactive
- FMGR 073 Account not found
- FMGR 074 JO command expected
- FMGR 075 Can't restore Type 6 PGM (user protected)
- FMGR 076 Can't restore Type 6 PGM (group protected)
- FMGR 077 Can't restore Type 6 PGM (insufficient capability)

FMGR UNNUMBERED

ERROR

MESSAGE MEANING

**ABEND
OPERATOR** The job has been aborted by operator request, or has been aborted because of spool I/O error.

**JOB xxxxx
ABORTED** Error encountered during job execution.

**ABEND EOJ
IN ssssss** An :EO or :JO command was encountered, but in a different level from the original :JO command. For example, control has transferred from PROG1 to PROG2. PROG2 contains :EO or :JO command. ssssss is the file name or logical unit number where :EO or :JO occurred.

**ABEND
JOB
LIMIT** The job time limit (set via the :JO command) has been exceeded.

**ABEND
RUN
LIMIT** The run time limit (set via the :TL command) has been exceeded.

**FMGR
WAITING
ON LU xx** LU xx is down locked.

FORMAT ERROR CODES

ERROR

CODE

EXPLANATION

- | | |
|----|---|
| 01 | a. w or d field does not contain proper digits.
b. No decimal point after w field.
c. $w - d \leq 4$ for E- specification. |
| 02 | a. FORMAT specifications are nested more than one level deep.
b. A FORMAT statement contains more right parentheses than left parentheses. |
| 03 | a. Illegal character in FORMAT statement.
b. Format repetition factor of zero.
c. FORMAT statement defines more character positions than possible for device.
d. List items remain and no conversion items are accessible in FORMAT statement. |
| 04 | Illegal character in fixed field input item or number not right-justified in field. |
| 05 | A number has an illegal form (e.g., two E's, two decimal points, two signs, etc.). |

FORTRAN ERROR CODES

ERROR CODE	EXPLANATION
01	Compiler control statement missing
02	Error in compiler control statement
03	Symbol table overflow
04	Labeled common
05	Implicit statement used to define default type for some character more than once
06	End of file occurred before "\$"
07	Return in main program
08	Illegal complex number
09	Mismatched or missing parenthesis
10	Illegal statement
11	Illegal decimal exponent
12	Integer constant exceeds maximum integer size
13	Hollerith string not terminated
14	Constant overflow or underflow
15	Illegal sign in logical expression
16	Illegal octal number
17	Missing operand — unexpected delimiter
18	Illegal constant usage
19	Integer constant required
20	Empty Hollerith string
21	Non-octal digit in octal constant
22	Illegal usage of name

- 23 Do terminator defined previous to do statement
- 24 Illegal constant
- 25 Illegal subprogram name usage
- 26 Integer variable or constant required
- 27 Statement number previously defined
- 28 Unexpected character
- 29 Only statement number on source line
- 30 Improper DO nesting or illegal DO terminating statement
- 31 Statement number starts with non-digit
- 32 Invalid statement number or illegal usage of a statement number
- 33 Variable name used as subroutine name
- 34 Statement out of order
- 35 No path to this statement or unnumbered format statement
- 36 Doubly defined common name
- 37 Illegal use of dummy variable
- 38 More subscripts than dimensions
- 39 Adjustable dimension is not a dummy parameter
- 40 Impossible equivalence group]
- 41 Illegal common block extension
- 42 Function has no parameters or array has empty declarator list
- 43 Program, function or subroutine or block data not first statement
- 44 Name in constant list in data statement

- 45 Illegal exponentiation
- 46 Function name unused or subroutine name used
- 47 Format specification not a local array name, statement number or * or it is an EMA reference
- 48 Illegal use of EMA
- 49 Improper use of name
- 50 DO statement in logical IF
- 51 Control variable repeated in DO nest
- 52 Logical IF within logical IF
- 53 Illegal expression or illegal delimiter
- 54 Doubly defined array name
- 55 Logical conversion illegal
- 56 Operator required logical operands
- 57 Operator requires arithmetic operands
- 58 Complex illegal
- 59 Incorrect number of arguments for subprogram
- 60 Argument mode error
- 61 Logical IF with three branches
- 62 Arithmetic IF with no branches
- 63 Required I/O list missing
- 64 Free field output illegal
- 65 Hollerith constant with count greater than 8 used in other than format or subprogram reference
- 66 Program unit has no body or block data subprogram has a body

- 67 Source file open or access problem or EOF, END\$ or \$ occurs before end statement
- 68 External name has more than five characters
- 69 Octal string in stop or pause statement is too long
- 70 Equivalence group syntax
- 71 Dummy variable in data list
- 72 Common variable in data list or in block data subprogram
- 73 Mixed mode in data statement
- 74 Illegal use of statement function name
- 75 Recursion illegal
- 76 Double defined dummy variable
- 77 Statement number ignored
- 78 Program unit has no executable statements
- 79 Format does not start with left parenthesis
- 80 Format does not end with right parenthesis
- 81 Illegal equivalence group separator
- 82 Illegal use of array name in an equivalence group
- 83 Subprogram name retyped
- 84 Object code memory overflow
- 85 Possible recursion may result
- 86 Dummy variable in statement function cannot be subscripted

88	End or format statement in logical IF
89	Continue statement or no branch in logical IF
90	First record of subprogram is a continuation line
91	Result of rename duplicates existing external name
92	Result of rename duplicates required intrinsic
93	Data statement attempts to initialize EMA variable
94	Name in EMA statement is not formal parameter or appears twice in the statement
96	A break was detected
97	Open or write error on binary file
98	Read access error on scratch file
99	Write access error on scratch file

The use of these names as program, subprogram, or common block names may result in a recursive operation if the program, subprogram, or common block contains an implicit call to a name that duplicates its own name (see error number 85).

ABS	CSGRT	DMAX1	LAND	TANH
AINI	CSIN	DMIN1	IFIX	
ALOG	DABS	DMOD	INT	
ALOG10	DATAN	DSIGN	IOR	
ALOGT	DATAN2	DSIN	ISIGN	
ATAN	DATN2	DSQRT	ISSW	
CCOS	DBLE	DTAN	NOT	
CEXP	DCOS	DTANH	REAL	
CLOG	DDINT	ERR0	SIGN	
CLRIO	DEXP	EXEC	SIN	
CMPLX	DLOG	EXP	SNGL	
CONJG	DLOG10	FLOAT	SQRT	
COS	DLOGT	IABS	TAN	

GASP ERROR CODES

GASP -33	Not enough room on cartridge
GASP -32	Cartridge not found
GASP -14	Directory full
GASP -13	Disc locked
GASP -12	EOF or SOF error
GASP -8	File open or lock rejected
GASP -7	Illegal security code or illegal write on lu2 or 3
GASP -6	File not found
GASP -4	More than 32767 records in a Type 2 file
GASP -2	Duplicate file name
GASP -1	Disc error, disc is down.
GASP 1	Disc associated with lu NN is down
GASP 2	Number out of range
GASP 3	Bad job number!
GASP 4	Illegal status
GASP 5	Illegal command
GASP 6	Not found
GASP 43	Lu not found in SST
GASP 46	Insufficient capability
GASP 55	Missing parameter
GASP 56	Bad parameter

I/O CALL ERROR CODES

IO00	An illegal class number was specified. Outside table, not allocated, or bad security code.
IO01	Not enough parameters were specified.
IO02	An illegal logical unit number was specified.
IO03	Illegal EQT referenced by lu in I/O call (select code=0).
IO04	An illegal user buffer was specified. Extends beyond RT/BG area or not enough system available memory to buffer the request.
IO05	An illegal disc track or sector was specified.
IO06	A reference was made to a protected track or to unassigned LG tracks.
IO07	The driver has rejected the call.
IO08	The specified disc transfer is longer than one track.
IO09	The LG tracks overflowed.
IO10	Class get call issued while one call already outstanding.
IO11	A Type 4 program made an unbuffered I/O request to a driver that did not do its own mapping.
IO12	An I/O request specified a logical unit not defined for use by this session.
IO13	An I/O request specified an lu which was either locked to another program, or pointed to an EQT which was locked to another program.

IO20	Read attempted on write only spool file.
IO21	Read attempted past end-of-file.
IO22	Second attempt to read JCL card from batch input file by other than FMGR. Revise program and re-run.
IO23	Write attempted on read only spool file.
IO24	Write attempted beyond end-of-file; usually, spool file overflow.
IO25	Attempt to access spool lu that is not currently set up.
IO26	I/O request made to a spool that has been terminated by the GASP KS command.
IOET	An end-of-tape condition occurred on the specified lu.
IONR	The specified lu is not ready. Make the device ready and set the EQT up.
IOTO	The specified lu has timed out.
IOPE	A parity error occurred in the data transmission from the specified lu.
ILL	INT an illegal interrupt occurred on the specified channel.

LIBRARY ERRORS

Mathematical Subroutines

OF = Integer or Floating Point Overflow

OR = Out of Range

UN = Floating Point Undefined

Error Message	Issuing Subroutine	Where Used	Error Condition
02-UN	ALOG	ALOG ALOGT CLOG	$X \leq 0$ $X \leq 0$ $x = 0$
03-UN	SQRT	SQRT DSQRT	$X < 0$
04-UN	.RTOR	.RTOR	$X = 0, Y \leq 0$ $X < 0, Y \neq 0$
05-OR	SIN	SIN CSNCS CEXP COS	$\left\{ \frac{1}{2} \left \frac{X}{\pi} + \frac{1}{2} \right > 2^{14} \right.$
06-UN	.RTOI	.RTOI	$X = 0, Y \leq 0$
07-OF	EXP	EXP CEXP .RTOR CSNCS	$X * \log_2 e \geq 124$ $X_1 * \log_2 e \geq 124$ $ X * \text{ALOG}(X) \geq 124$ $X_2 * \log_2 e \geq 124$
08-UN	.ITOI	.ITOI	$I = 0, J \leq 0$
08-OF	.ITOI	.ITOI	$I^J \geq 2^{15}$ or $I^J < -2^{15}$
09-OR	TAN	TAN	$X > 2^{14}$
10-OF	DEXP	DEXP .DTOD .DTOR .RTOD	$e^X > (1-2^{-39}) 2^{127}$ $\left\{ X > (1-2^{-39}) 2^{127} \right.$

11-UN	DLOG	DLOG	$X \leq 0$
		DLOGT	$X < 0$
12-UN	.DTOI	.DTOI	$X = 0, I \leq 0$
13-UN	.DTOD	.DTOD	$X = 0, Y \leq 0$
		.DTOR	$X < 0, Y = 0$
		.RTOD	
14-UN	.CTOI	.CTOI	$X = 0, I \leq 0$
15-UN	DATN2	DATN2	$X = Y = 0$

Utility Subroutines

Subroutine	Error
MAGTP	Returns on an illegal call.
.SWCH	Returns if element is out of range.

LOADR ERROR CODES

C-CK SUM

L 01 This is a checksum error. Most likely you specified a file to the LOADR that did not contain relocatable format code.

L-IL REC

L 02 The LOADR found a record that was not a NAM, ENT, EXT, DBL, EMA, or END record.

L-OV MEM

L 03 The size of the code loaded so far exceeds the max size that you specified or exceeds the largest possible size for a program.

L-OV BASE

L 04 Base page overflow. This program has used too many base page links.

L-OV SYM

L 04 This is a symbol table overflow.

L-CM BLK

L 06 This is a common block error.

L-DU ENT

L 07 Duplicate entry point.

L-TR ADD

L 08 No transfer address. Only subroutines were loaded.

L-RE SEQ

L 09 Record out of sequence.

L-IL PRM

L 10 The run string submitted to the LOADR was in error.

L-CO RES

L 11 Attempt to replace a memory resident program.

L-OV FIX

L 12 Fixup table overflow.

L-LM LIB

L 13 The limit on the number of libraries specified by the 'LI' command has been exceeded (10).

L-IL REL

L 14 The compiler produced an illegal record.
Recompile.

L-IL PTN

L 16 You specified a partition in the load of the program, however, that partition does not exist or has been downed due to a parity error.

L-RQ PGS

L 17 The number of pages that you specified in the load of the program exceeds that number of pages in the partition you specified.

L-OV PTN

L 18 The specified program size is too large for the partition.

L-ML EMA

L 19 Illegal EMA declaration.

L-ID EXT

L 20 No ID extensions available for the EMA program.

L-SZ EMA

L 21 The programs declared EMA size is too large for this systems partition definitions.

L-SS ENT

L 24 You attempted to access an SSGA entry point but you did not 'OP,SS'.

L-IL CMD

L 25 Attempt to purge a program under batch or attempt to use the 'LI' or 'PU' commands within a LOADR command file.

L-ID SEG

L 26 Not enough short and long ID segments to finish the load.

L-RF EMA

L 27 Attempt to access an EMA external with offset or indirect.

L-UN EXT

L 28 Undefined externals exist which prohibits the load from completing.

L-EX CPY

L 29 Attempt to replace or purge a program where copies of that program exist.

L-RP CPY

L 30 Attempt to replace a copied program.

L-PE LDR

L 31 Trying to do a purge or permanent load with a copy of the LOADR.

L-DU PGM

L 32 You tried to load the same program several times but did not remove the earlier loads.

L-NO IDS

L 33 Not enough ID segments to finish the load.

L-RP PGM

L 34 You tried to replace a permanent program.

LOGON ERROR CODES

- LGON 06** this is an informational diagnostic. The station (terminal) being logged onto has a configuration table entry which is a duplicate of an entry in the users account file entry.
- LGON 09** Your session has exceeded the maximum session switch table size.
- LGON 11** The LOGON program received the specified error when attempting to mount a private or group disc to this session.
- LGON 13** LOGON detected a user SST which attempted to redefine a system disc's logical unit number.

LU LOCK ERROR CODES

- LU01** A program has one or more logical units locked and is trying to lock another with wait.
- LU02** Illegal logical unit reference.
- LU03** Not enough parameters are furnished in the call.
- LU04** Trying to lock a logical unit not defined in caller's SST.

OUTSPOOL ERROR MESSAGES

MESSAGE	CAUSE
JOB WAIT ON PT	End-of-Tape occurred between :JO and :EO commands.
JOB WAIT ON SPOOL RESOURCE	Required spool file or logical device cannot be obtained at this time.
JOB WAIT ON EXTENT	Spool file overflows available disc space.
END JOB ABNORM	JOBFIL could not be opened; or other uncorrectable error occurred; or JOB was run before spool initialization.
BAD EOF	Message appears after last line of file. ASCII file outspooling overflowed; or was otherwise incomplete.

READT/WRITT ERROR CODES

- | | |
|-----------------|--|
| READ 001 | The requested mag tape unit is down. |
| READ 002 | The mag tape READT is trying to restore contains information in a format not restorable by READT. |
| READ 003 | The mag tape unit you wish to use is locked to some process. |
| READ 004 | The parameter describing the desired mag tape unit does not satisfy READT's requirements for a legal mag tape lu. |
| READ 005 | The desired mag tape unit is off-line. |
| READ 006 | READT rejected the use of the specified disc lu. |
| READ 007 | The driver detected a parity error when reading from the mag tape. |
| READ 008 | The end of tape was reached. |
| READ 009 | The desired cartridge has a file open or the cartridge is locked to another program. |
| READ 010 | You are operating in a nonsession environment. An lu must be specified (negative lu) since there isn't a free disc pool. |
| READ 011 | READT rejected the size (number of tracks) you specified. |
| READ 012 | The routine READT uses to mount a cartridge detected an error. |
| READ 013 | The desired disc lu or the available free lus in the disc pool are not large enough to restore the cartridge that's on the mag tape. |

- READ 014** The FMP tracks on lu 2 or lu 3 (if 3 exists) are not restorable with READT.
- WRIT 001** The device can be enabled.
- WRIT 002** Only the system manager can save system discs.
- WRIT 003** The mag tape you wish to use is locked to some process.
- WRIT 004** The parameter describing the desired mag tape unit does not satisfy READT's requirements for a legal mag tape unit.
- WRIT 005** The desired mag tape unit is off-line.
- WRIT 006** A write ring is required to write information on a mag tape.
- WRIT 007** The driver detected a parity error when reading from the mag tape.
- WRIT 008** The end of tape was reached.
- WRIT 009** The desired cartridge has a file open or the cartridge is locked to another program.
- WRIT 010** The desired cartridge or disc lu could not be found.
- WRIT 011** WRITT rejected the use of the specified disc lu.
- WRIT 012** You cannot save FMP tracks off lu 2 or lu 3 (if 3 exists) with WRITT.

RECONFIGURATION ERROR CODES

CONFIG

ERR

MEANING

- | | |
|----|--|
| 1 | Invalid LU number or a bit bucket LU. |
| 2 | Illegal select code number. |
| 3 | New select code entered is identical to new select code assigned to disc system console or list device, or else the current select code entered is identical to the old select code for disc, system console or list device (i.e., do not reconfigure that which was already done via the SWTCH register). |
| 10 | Specified total number of pages outside the range. |
| 11 | Invalid bad page number. |
| 12 | Specified SAM extension entry beyond physical memory size due to bad pages. |
| 13 | Current running total exceeds available pages in block of good memory or exceeds size of mother partition. |
| 14 | Second parameter of partition definition entry other than RT, BG or S, or else S was entered when a subpartition definition was not expected. |
| 15 | Third parameter of partition definition entry other than R. |
| 16 | No such program, or the name of a segment was entered or invalid type was entered for partition assignment. |

- | | |
|----|---|
| 17 | Invalid partition number. |
| 18 | Program does not fit in the assigned partition. |
| 19 | Invalid number of pages was entered for program size. |
| 20 | Number of defined partitions already equal to allowed maximum number and more undefined pages remain. |
| 21 | Page requirements of an EMA program cannot be modified. |
| 22 | Number of pages in SAM extension requires division into more than five blocks. |

RESOURCE NUMBER ERRORS

- | | |
|-------------|--|
| RN00 | There are no option bits set in the call. |
| RN01 | Not used |
| RN02 | The specified resource number is not defined. |
| RN03 | An unauthorized attempt was made to clear a local resource number. |

SCHEDULE CALL ERROR CODES

SC00	A batch program attempted to suspend (EXEC(7)).
SC01	Missing parameter.
SC02	Illegal parameter.
SC03	The specified program cannot be scheduled.
SC04	The specified program is not a subordinate (or "SON") to the program issuing the completion call.
SC05	The program given is not defined.
SC06	No resolution code is specified in the execution time EXEC call.
SC07	A prohibited core lock was attempted.
SC08	The program just scheduled is assigned to a partition smaller than the program itself or to an undefined partition.
SC09	The program just scheduled is too large for any partition of the same type.
SC10	There is not enough system available memory for the string passage.
SC11	EXEC schedule or timed execution request was issued and program specified is already in the time list for another session.

SMP ERROR MESSAGES

ERROR

MESSAGE MEANING

SMP:LU xx File filename just outspooled to logical
EOFER unit xx overflowed or was otherwise
filename incomplete.

SMP:LU xx Logical unit xx down: filename placed in
DOWN active hold.
filename
HELD

SMP:FMP FMP error —nn occurred during SMP
—nn operation. Usually indicates loss of JOB-
 FIL of SPLCON.

SYSTEM AND BREAK-MODE COMMAND ERROR MESSAGES

ERROR MESSAGE	MEANING
------------------	---------

OP CODE ERROR	Illegal operator request code.
------------------	--------------------------------

NO SUCH PROG	The name entered is not a main program in the system.
-----------------	---

INPUT ERROR	A parameter is illegal.
----------------	-------------------------

ILLEGAL STATUS	Program is already scheduled.
-------------------	-------------------------------

CMD IGNORED — NO MEM	Not enough system available memory exists for storing the program's command string.
----------------------------	---

ILLEGAL PART'N	Partition does not match command request.
-------------------	---

SIZE ERROR	Illegal program size specified or size of program specified larger than its assigned partition or any partition.
---------------	--

SYSTEM BOOT-UP HALTS (front panel)

HLT

MEANING

- | | |
|-----|---|
| 4 | Powerfail occurred and powerfail automatic restart is enabled. |
| 5 | Memory protect switch was set and memory parity error occurred. |
| 10B | FMGR or D.RTR cannot be scheduled at startup because there is not a large enough partition (issued by the system). |
| 11B | Attempt was made to re-execute a non-RPL compatible ROM Loader Part # 12992A, or Bootstrap Loader. |
| 22B | SCNFG cannot find an ID segment for Configurator extension \$CNFX, \$CNFX is not a Type 3 program, or a contiguous memory block of three good pages cannot be found in the user partition area. |
| 30B | Error was encountered in the disc I/O process by one of the RPL-compatible ROM Loaders Part # 12992B and 12992F. If the disc is a 7900 the disc status is displayed in the A-register. If the disc is a 7905/20 the disc status word 1 is displayed in the B-register and disc status word 2 in the A-register. |
| 31B | Error encountered in the disc I/O process by the Boot Extension. If the disc is a 7900, the disc status is displayed in the A-register. If the disc is 7905/06(H)/20(H)/25(H), the disc status word 1 is displayed in the B-register and disc status word 2 is displayed in the A-register. |
| 55B | An EQT with the equipment type code of console cannot be found. |



**HEWLETT
PACKARD**

**DATA SYSTEMS DIVISION
11000 WOLFE ROAD
CUPERTINO, CALIFORNIA 95014**